IN THE UNITED STATES PATENT & TRADEMARK OFFICE

APPLICANT: MICHAEL E. TOMPKINS ET AL	§	GROUP ART UNIT: 2314
SERIAL NO.: 08/162420	§	
FILING DATE: DECEMBER 3, 1993	§	
FOR: SPA CONTROL SYSTEM	§	EXAMINER: E. RAMIREZ

SECOND DECLARATION OF MICHAEL E. TOMPKINS UNDER 37 C.F.R. 1.132

Commissioner of Patents & Trademarks
Washington, D.C. 20231

Date: May 9, 1994
Docket No.: 86-1198-00

Sir:

Applicant states as follows:

1. My name is Michael E. Tompkins and I am an inventor of the above-referenced application. I am over the age of twenty-one, have never been convicted of a felony, and am competent to give this Declaration. The following statements are of my own personal knowledge.

2. I have read a copy of the Examiner's Office Action dated March 10, 1994.

3. In the late 1970's several companies manufactured solid state microcomputer control products that were designed for industrial control applications. Their primary markets were the applications that utilized electro-mechanical (relay) components for logic and control. These solid state microcomputer-based products, commonly known as Programmable Logic Controllers (PLC), are designed with the very same type of hardware components listed in the above-referenced application. For reference, Exhibit A contains the datasheet for a PLC system that was introduced by Gould in 1982. Because these PLC systems were designed for relay control system replacements, and were generally available to the public, anyone actively engaged in the design of electro-mechanical control systems, including for spa control, would have had PLCs and their literature available to them.

4. Although these PLC systems contained a microcomputer, no knowledge of design or programming of a computer is required for their implementation. The specific details of the internal microcomputer "program" are not required for their "configuration"

to meet the specific application of the PLC to the control problem to be solved, even though the details of the "program" are well documented for those interested in its functions, as shown in literature in Exhibits A, C, E and F.

5. The term "programming" is being mis-used and/or used inconsistently by the Office. It is not necessary to be able to "program" a microcomputer in any language to perform the specific functions listed in the above-referenced application, although it is possible to do this based on the literature of Exhibits A, C, D and F. The only requirements are that; 1) the control system and its hardware must be detailed as described in Figures 1,2 and 3 in the above-referenced application, and 2) a configuration must be entered into the PLC that reflects the algorithms described in the above-referenced application along with Figure 4. The application used the term "program" to indicate the specific control configuration to be implemented by any of a variety of control devices that form further program support for the control algorithms, into which the control strategy "program" could be loaded.

6. Someone skilled in the art of designing spa control systems with electro-mechanical devices would have been able to buy the product in Exhibit A, or similar products such as shown in Exhibit F, well before May 27, 1987. Such person could also "configure" it to perform all the control and measurement functions set forth in this application to provide the solution described in the above-referenced application to the control requirements of spas, identified in the above-referenced application, if the person skilled in the art, had access to the above-referenced application. This implementation of the solid-state PLC uses the same fundamental system design techniques previously used in creating electro-mechanical relay logic technology, without undue experimentation.

7. The uniqueness of the invention of the above-referenced application lies in recognizing the control problem and formulating the solution to it that are detailed in the above-referenced

application. The information provided in the above-referenced application would have enabled someone skilled in the art to select the specific hardware required from the catalogs of various PLC manufacturers, such as that of Exhibits A and F, and to configure and connect them, as in Figures 1 and 3 of the above-referenced application, to perform all the operations of the invention before May 27, 1987.

8. To utilize a PLC in the above-referenced application, one skilled in the art, would review the requirements of the above-referenced application and create a block diagram of the product similar to that in Figure 2 of the application, separating the design into the actual inputs and outputs, assuming at this point that the PLC is a generic control device as depicted in the literature, such as Exhibits A and F. Next, the inputs and outputs would be separated accordingly into specific characteristic groups such as, digital, analog, high voltage, high current, etc. This would create the block diagram shown in Exhibit B page 1. Then, using the data provided by the PLC manufacturer, such as that in Exhibit A page 14, the details of the inputs and outputs would be matched to the list of modules available. For example, there are 13 digital inputs in this application which are all low voltage DC signals which could connect directly to the Gould Model B837-016, a digital input module with 16 inputs. The other modules to accept the analog inputs and digital outputs would then be selected accordingly for the application. This would now define the hardware components required for the product and provide a system block diagram as shown in Exhibit B page 3. The primary PLC hardware for this system, based on the Gould Model 884 PLC shown in Exhibit A, would then consist of the following components:

- Model 884A-101 controller;
- Model P884-001 Power Supply Module;
- Model B837-016 16 channel digital input module;
- Model B824-016 16 channel digital output modules;
- Model B873-001 4 channel analog input module;
- Model B884-002 Proportional, Integral and Derivative module.

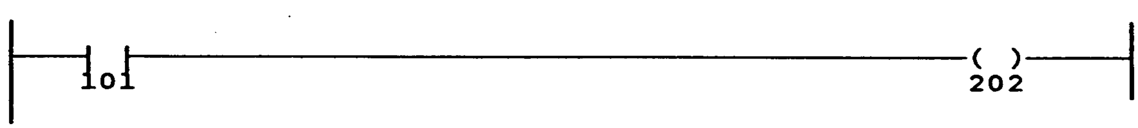
This configuration is substantially identical to Figure 2 of the above-referenced application. This system would have been easy to configure, from the control literature, and would perform all of

60

the functions of the system that is described in the above-referenced application.

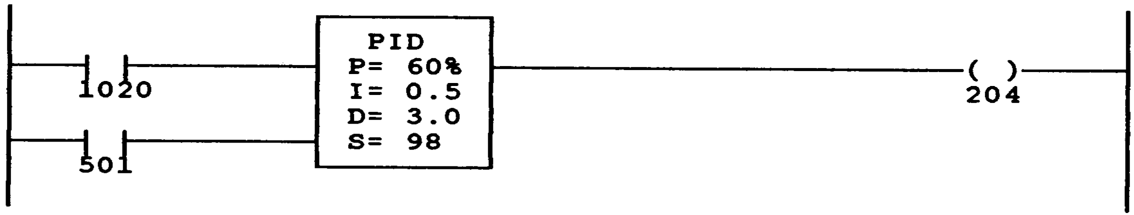
9. PLCs can be configured or "programmed" in several methods, but the method normally provided by the manufacturer is Relay Ladder Logic. This form of "configuration" allows those skilled in the art of control system design, using electro-mechanical relays, to easily configure and operate these solid state devices. The PLC systems were designed to allow electricians, maintenance personnel and system designers to install and utilize them without requiring extensive training or undue experimentation. The only training required would be on the method of configuration of the PLC which may vary slightly from one manufacturer to another and is reflected in the user's manuals. Pertinent portions of a PLC users manual are attached in Exhibit D.

10. The configuration of the PLC would then continue by extracting the functions required in the application and then converting these functions into relay logic form which is the configuration language or "programming language" of the PLC. Each input and output is assigned a unique address determined by where it is physically wired. These addresses are then made a part of the configuration to identify the physical inputs and outputs. For example, if the Jets switch input is wired to the first point of the digital input module, it has an assigned address of 101. If the pump motor for the jets is connected to the second output of the digital output module it will have an address of 202. After a map is made of the physical inputs and outputs ("I/O"), then the functional description in the above-referenced application is turned into a ladder logic program. In this example, if the Jets pump output is to be activated when the Jets switch is activated, a relay ladder logic rung would be created similar to the following:



61

This indicates that when the normally open contact of digital input 101 (the Jets switch) is activated that the output device 202 (the pump motor output) will be energized. Complex functions like the PID control of the heater would be handled in a similar manner.



This ladder rung indicates the operation of the PID control algorithm. When the control relay 1020 closes to allow power to the PID control algorithm, and the temperature sensor connected to analog input 501 is lower than the setpoint, set by the operator's control, the module will activate the heater output 204. When the PID module determines that the water temperature has reached the setpoint, it will deactivate the 204 output, turning off the heater. The parameters shown in the PID box with particular values are solely for illustrative purposes. The actual values that determine the rate at which the on/off heating cycle occurs, also known as "tuning the loop", are selected dependent upon the physical equipment used and the temperature accuracy required, as anyone skilled in the art would know. The rest of the functions of the spa application are configured in a similar manner.
Exhibit C contains relay ladder logic configuration examples and explanations, similar to those above, taken from the Gould 484 Applications Manual, which was available before May 27, 1987. Included in these examples are; alarm point monitoring, conversion algorithms, mathematical calculations, scan programs, etc.

11. Exhibit D contains sections from the Gould Model 584 PLC user's guide, (available in 1978), that explain the internal operations and the internal "program" of the microcomputer for I/O scanning- pages 29 and 61, memory utilization- page 63, and data base operation- page 64. In several sections of the user's guide, and specifically on page 61 paragraph 3, statements similar to the

following are given: "This information is NOT required for the designer to program the controller".

12. Exhibit E contains the datasheets and descriptions of PLC interface modules, manufactured by the Allen-Bradley Company, that perform the functions of; digital I/O, analog I/O, PID control and real-time clock. These modules were available before May 27, 1987.

13. As further evidence of the documentation required to build this invention, Exhibit G contains a copy of the original design specification, dated January 2, 1987, from which the prototype of the product was built. This documentation provides less detail to the designer than the patent application. The product was designed from this document using principals generally known in the art for configuring a generalized control system, in conjunction with the design specification, without undue experimentation in a period of approximately four months by persons not knowledgeable specifically in spa control and design, but were generally knowledgeable of control systems and microcomputer design applications. The computer system built for this application was a custom designed special purpose controller and not a PLC. If a pre-programmed unit such as the PLC described in this declaration had been used, the prototype could have been made in a much shorter period of time.

14. The above-referenced application provides the specific details of the final application and is not a tutorial on the art of combining microcomputers, well known in the art, and programs at that time well known in the art, to provide a control system, as described in my previous Declaration and the other previous Declaration, both filed in the above-referenced application. The block diagrams and internal operations only reference the specific components used in the control application described in the above-referenced application, and specific detail was not required as the functions, such as input scanning, alarms, control, A-D conversion are well known in the field, and are well documented, and are included in many industrial control systems, such as the PLC units

referenced in this Declaration. As stated in my previous Declaration, any computing device of the period beginning in the late 1970's, with appropriate input and output hardware as set out in the above-referenced application, could have been used to perform and execute the listed operations and algorithms described in the above-referenced application specifications and figures. The implementation of the hardware and the control functions are unique, not the specific hardware and general scan control and alarm programs that facilitate the performing of the unique functions of the invention.

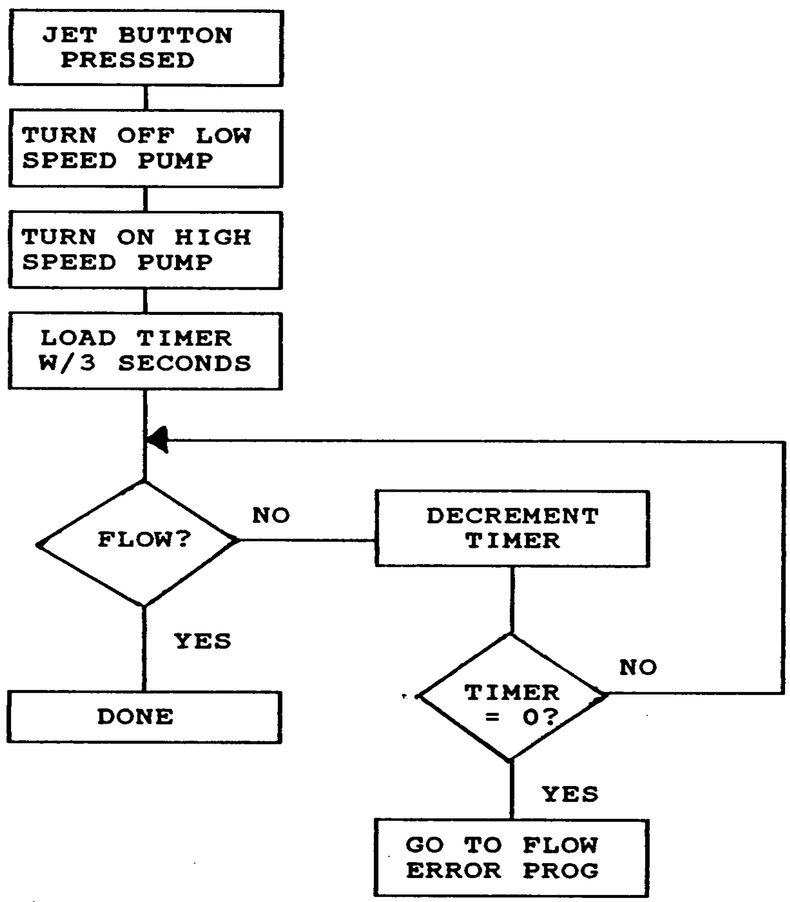
15. Exhibit F contains the literature for several other PLC manufacturers and their products that were available before May 27, 1987 that also were considered to be direct electro-mechanical replacement systems and could have been used to implement the invention of the above-referenced application.

16. The above-referenced application contains operational descriptions of the invention that are sufficient to allow someone knowledgeable in the art of control system design to create flowcharts, or other pre-programming diagrams, that could be used to construct a control system program or configuration. This program or configuration can then be tailored specifically to match the unique characteristics of the language and the control system hardware of their choice. While visually, the flowcharts may be easier for some people to understand, the same technical information is contained in the body of the text. This is not unlike the comparison of a mathematical word problem to an equation. Someone skilled in the art of control system design, as part of their normal job functions, should be able to create a flowchart from written or verbal descriptions.

For example, the following text taken from page 7 line 15 of the application, that describes the operation of the Jets button;

"The jet button operates the high speed pump for the jet action in the spa. After the jet button is depressed, the system will shut off the pump if there is no flow in the system after three seconds of operation. The user is notified of the malfunction by an error message shown on the display. In a preferred embodiment, the low speed pump automatically is operated when the heater is activated. By pressing the jet button, the high speed overrides the low pump."

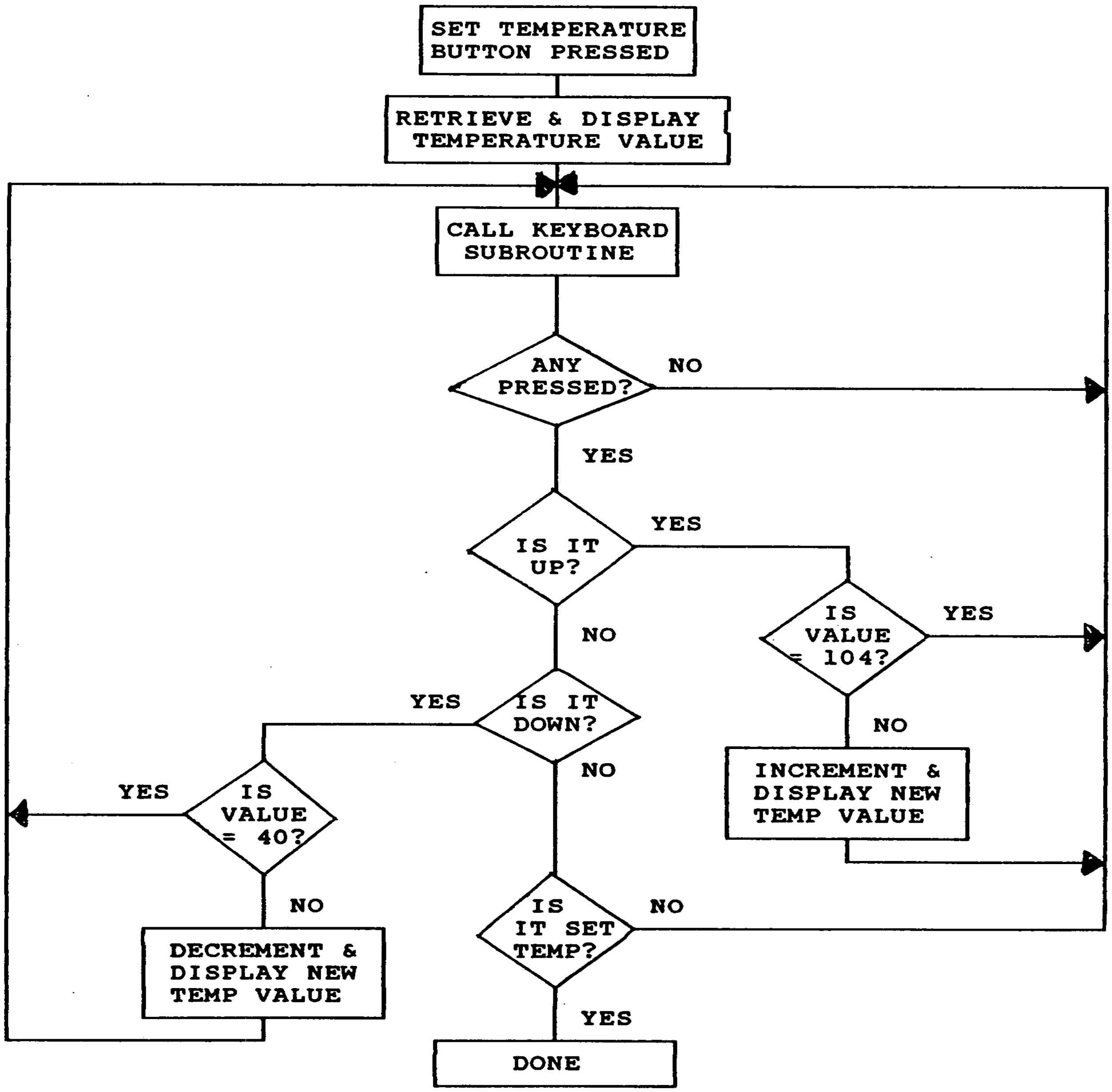
can be easily transformed directly into the following flowchart:



A second example, taken from page 7 line 35 of the application, that describes a more complex operation of setting the temperature;

"The set temperature button can be used to control the temperature value for the thermostat in the controller. To set the temperature, the set temperature button is depressed and the current setting for the thermostat will be shown on the display. The up arrow or the down arrow button can be used to increase or decrease the temperature setting as desired. When the desired value is shown on the display, the set temperature button is depressed and the system will revert to the normal scroll in the display. The ranges on the temperature setting may range from 40 to 104 degrees fahrenheit."

is also easily transformed into the following flowchart:



The other control functions described in the application are also transformed to flowcharts just as easily.

17. Certain descriptions of control operations presented in the above-referenced application are provided to describe a control system function that is generic and most likely specific to the control system designer. The exact details on these operations are not pertinent to the understanding of the invention. For example, the above-referenced application did not elaborate on the internal components and operation of the power supply for the invention, because, it is understood and accepted, by those skilled in the art, that the device transforms AC line power into a regulated DC output. As another example, the following statement taken from page 6 line 19 of the above-referenced application;

"The analog conversion Program manipulates the converter circuitry to convert sensor input signals to digital information."

does not require further description because the specific details of the actual conversion of analog signals into digital data is dependent on the control system hardware used, and in the case of the previously discussed PLC, is an internal operation that is performed transparent to the system designer, (see pages 3-5 and 3-6 of Exhibit E). The exact details of the conversion process performed is not pertinent to the overall function described.

18. I hereby declare that all statements made herein of my own knowledge are true and that all statements on information and belief are believed to be true; and further that these statements were made with the knowledge that willful, false statements and the like so-made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful, false statements may jeopardize the validity of the above-referenced application or any patent issued thereon.

Michael E. Tompkins

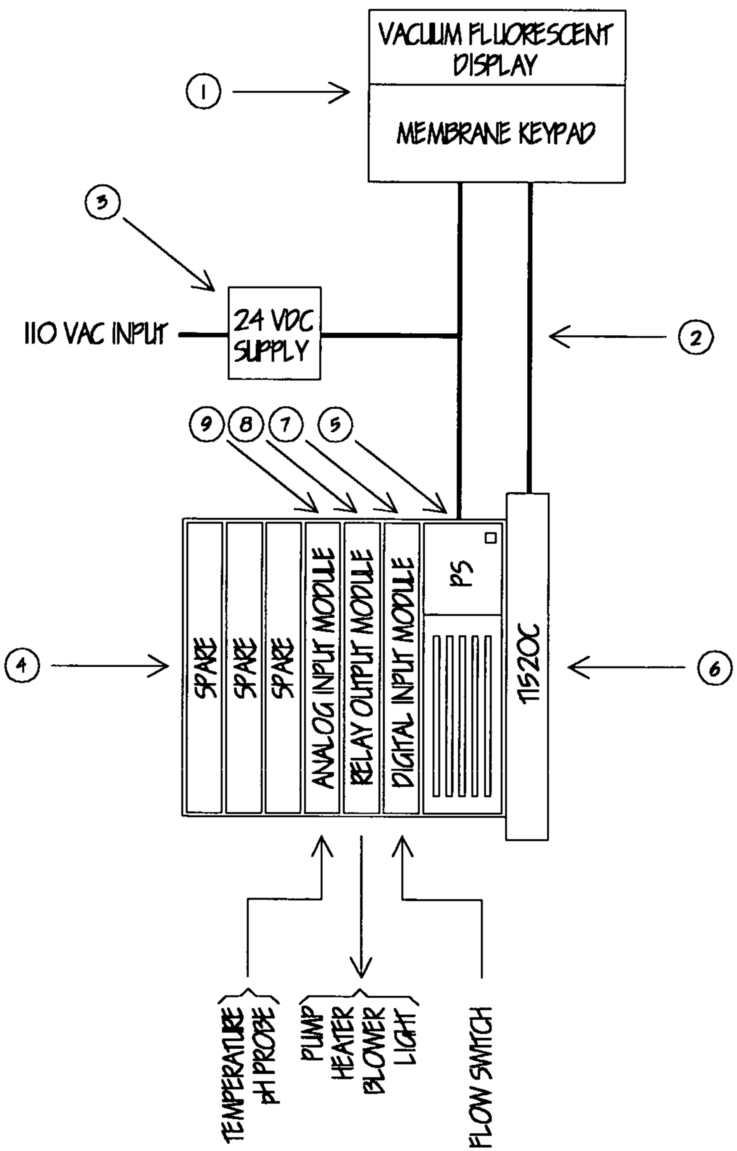
May 9, 1994

Date

PLC BASED SPA CONTROL SYSTEM DIAGRAM

S22

A



BILL OF MATERIAL

ITEM NO.	QTY.	DESCRIPTION	MANUFACTURER	PART NO.
1	1	OPERATOR INTERFACE	ICS DEVELOPMENT	ICSD-2210
2	1	RS-232 COMMUNICATIONS CABLE	ICS DEVELOPMENT	
3	1	24 VDC POWER SUPPLY	SOLA	
4	1	6-SLOT I/O BASE	TEXAS INSTRUMENTS	500-5892
5	1	24 VDC BASE POWER SUPPLY	TEXAS INSTRUMENTS	500-2153
6	1	TI520C PLC W/ 2K WORD OF MEMORY	TEXAS INSTRUMENTS	520C-1102
7	1	8-POINT, 24 VDC DIGITAL INPUT MODULE	TEXAS INSTRUMENTS	500-5008
8	1	8-POINT RELAY OUTPUT MODULE	TEXAS INSTRUMENTS	500-5061
9	1	8-CHANNEL ANALOG INPUT MODULE	TEXAS INSTRUMENTS	500-5037

B

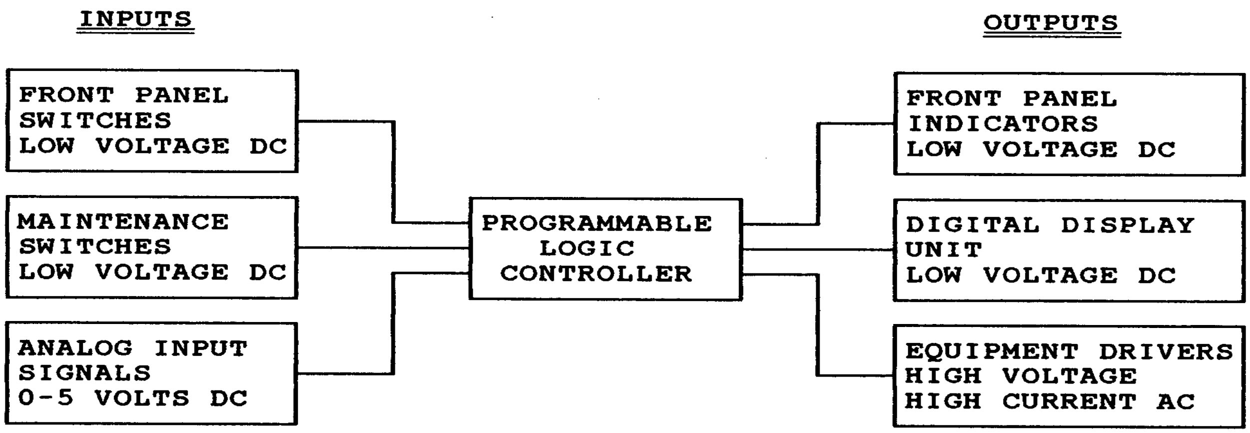

INPUTS
OUTPUTS
FRONT PANEL SWITCHES LOW VOLTAGE DC
MAINTENANCE SWITCHES LOW VOLTAGE DC
ANALOG INPUT SIGNALS 0-5 VOLTS DC
PROGRAMMABLE LOGIC CONTROLLER
FRONT PANEL INDICATORS LOW VOLTAGE DC
DIGITAL DISPLAY UNIT LOW VOLTAGE DC
EQUIPMENT DRIVERS HIGH VOLTAGE HIGH CURRENT AC

EXHIBIT B

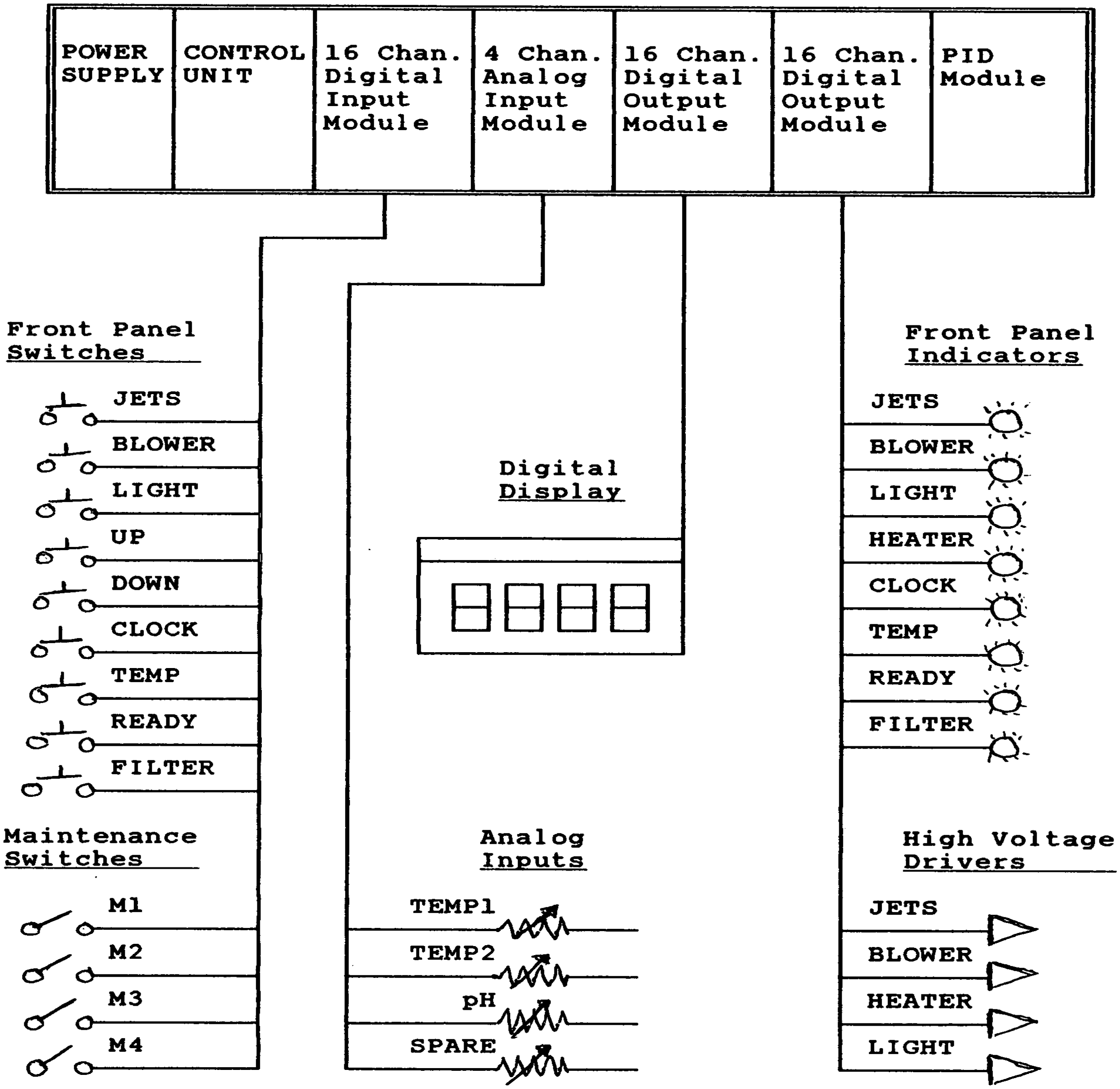

POWER SUPPLY
CONTROL UNIT
16 Chan. Digital Input Module
4 Chan. Analog Input Module
16 Chan. Digital Output Module
16 Chan. Digital Output Module
PID Module
Front Panel Switches
JETS
BLOWER
LIGHT
UP
DOWN
CLOCK
TEMP
READY
FILTER
Maintenance Switches
M1
M2
M3
M4
Digital Display
Analog Inputs
TEMP1
TEMP2
pH
SPARE
Front Panel Indicators
JETS
BLOWER
LIGHT
HEATER
CLOCK
TEMP
READY
FILTER
High Voltage Drivers
JETS
BLOWER
HEATER
LIGHT

EXHIBIT D

ML-584A-000 Rev D

Gould
584 Programmable Controller

USER'S GUIDE

GOULD
Electronics

447

over →

PREFACE

The MODICON 584 PC User's Manual provides the information necessary to configure and operate MODICON's 584 Programmable Controller. The manual describes system requirements, basic programming, internal functions, and installation. The programming sections assume that the user has a prior knowledge of the relay ladder diagram language.

Rev. B., January 1982

448

MODICON 584PC USER'S MANUAL
TABLE OF CONTENTS

449 over →

SECTION IV
DATA TRANSFER (DX) PROGRAMMING

SECTION V
INTERNAL PRINCIPLES

SECTION VI
INSTALLATION

APPENDICES

584 MANUAL-FIGURES

450

FIG. NO.

451 over

TABLES

451

Section I — Introduction

1.0 General

The MODICON 584 Programmable Controller User's Manual provides the information necessary to operate MODICON's 584 Programmable Controller. The manual describes system requirements, basic programming, internal functions, and installation. It is assumed that the user has some prior knowledge of the relay ladder diagram language.

The 584 Programmable Controller is a solid-state device designed to perform logic, timing, sequencing, and various calculations for industrial control applications. The 584 PC is general purpose and is used as a direct replacement for relays or solid-state electronics in an industrial environment. Its design makes it applicable for a wide variety of industries. These include:

Machine Tools	Plastics
Food Processing	Refinaries
Pipelines	Test Stands
Transfer Lines	Petrochemicals
Energy Management	Forest Products
Pulp and Paper	and many others

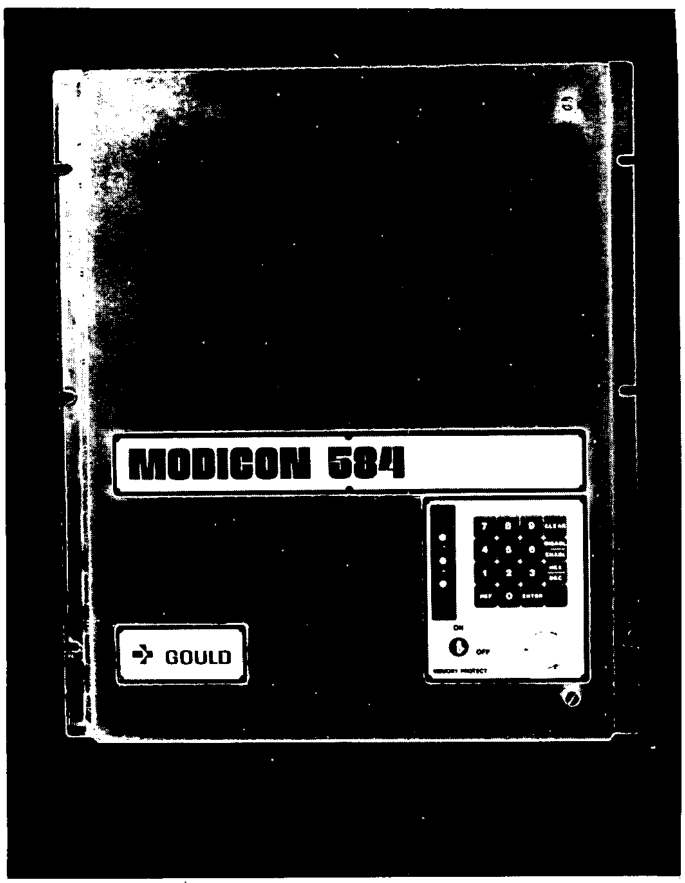


FIGURE 1 584 PROGRAMMABLE CONTROLLER, (PC)

MODICON's 584 PC is widely used to replace mini-computers in industry. When compared to these devices, PC's offer the following advantages:

- Solid-state construction for maximum reliability.
- Designed to operate in a hostile industrial environment (i.e. heat, electrical transients, EMI, vibration, etc.) without fans, air conditioning, or electrical filtering.
- Programmed in a simple ladder diagram language. New language need not be learned.
- Easily programmed with a portable CRT programming panel.
- Controller is reusable if no longer required on original application.
- Indicator lights provided at major diagnostic points to simplify troubleshooting.
- Simple maintenance, based upon module replacement, insures minimum downtime and maximum productivity.

The 584 Controller offers unparalleled flexibility, expanding from a 50 relay system to over 2000 relay equivalence. MODICON customers can select either a basic relay replacement system, a sophisticated diagnostic monitoring, data collection/storage, or a report generating system, as well as anything in between. These widely differing capabilities are provided with the same hardware to minimize the quantity of spare parts in large multiple controller installations. In addition to the advantages previously discussed, the 584 offers the following benefits:

- Low cost hardware costs less than installed relays.
- Flexible, start small and add data management later.
- Expandable Memory from 4K to 32K (16 bit words).
- Fast Scan Rate between 10 and 60 mSec.
- Uses existing MODICON I/O devices, fewer I/O spares required.
- Easy installation of field wiring, intermixing any type of I/O.
- Retentive memory for logic and numerical storage.
- Programming and peripheral devices plug directly into controller without effecting scan.
- Real-time, complete on-line programming, a MODICON standard of excellence for maximum flexibility.

A typical programmable controller can be divided into three components as shown in Figure 2. These components are the processor, power supply, and an input/output section. In the 584 PC, both the processor and the power supply are contained within the same cabinet.

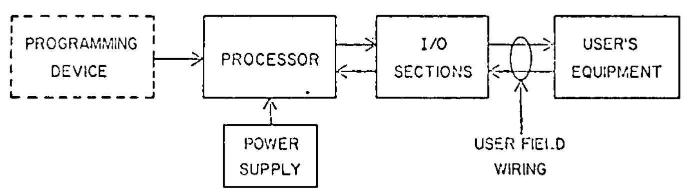


FIGURE 2 TYPICAL PC BLOCK DIAGRAM

1.1 Processor

The controller's processor is a complete solid-state device created by combining three unique printed circuit boards (see Figure 3). Each printed circuit board is mounted in its own chamber within the mainframe. Proceeding from left to right, the I/O processor is used basically to communicate to the I/O section. It also contains the controller's fixed intelligence. The second printed circuit board is the central processor which makes all the logical decisions for the controller. The final circuit is the memory board where system parameters, programmed logic, and numerical values are stored.

453 over

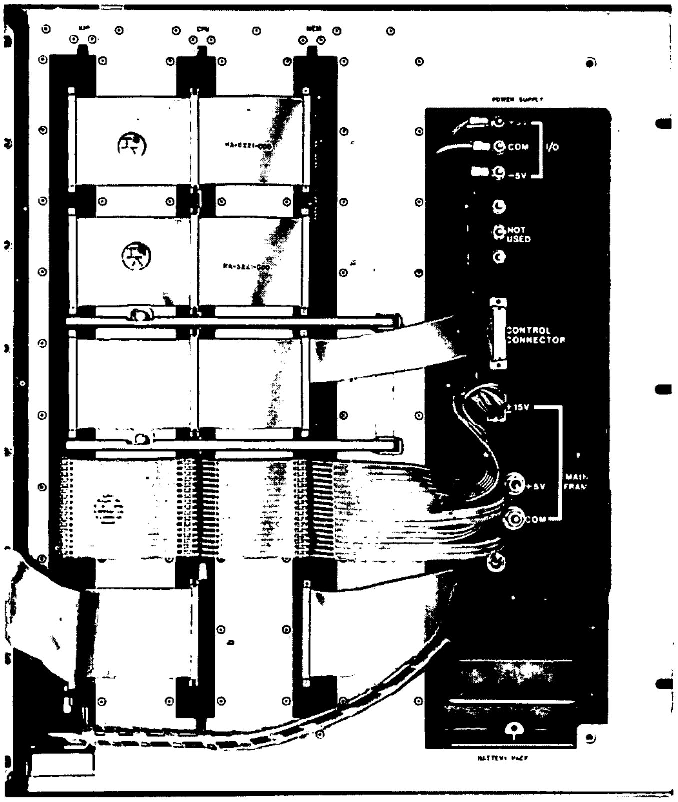


FIGURE 3 MAINFRAME INTERNAL COMPONENTS

The processor operates on DC power supplied by the power supply. This internal DC power is also routed through the processor to operate the I/O section. Once the ladder diagram program is entered into the processor, it remains intact until deliberately changed by the user with one of the programming devices. The program remains unaltered in the event of power failure or power off conditions.

An access port on the mainframe permits connection to the processor and can be used for entering instructions and data or monitoring previously entered information. The most common method of entering data into the processor is with the P190 CRT Programmer (Figure 4). Other devices that could also be connected to this port are a tape loader, a computer, or a telephone interface. A second access port located on the I/O processor has the same capabilities as the mainframe port. By using both ports, two external devices can be communicating to the 584 at the same time.

FIGURE 4 P190 PROGRAMMER

1.2 Power Supply

The largest, right most chamber in the mainframe contains the controller power supply. This power supply operates on AC power to produce DC power required for the controller's internal operation. It is designed to utilize either 115 VAC or 220 VAC (jumper selectable), as well as a wide frequency range (47-63 Hz). No major configuration changes are required other than a jumper adjustment to convert from 115V, 60 Hz operation to 220V, 50 Hz. However, different model power supplies are only compatible with certain memory types (CMOS or Core).

Adjustments or routine maintenance to the power supply is not required. Indication of operational power ready status is provided on the mainframe itself. No external cooling is required; however, free air circulation around the mainframe should be provided. The power supply has sufficient capacity to operate the processor, peripheral devices, and two channels of local I/O (maximum total 256 inputs and 256 outputs). Additional power supplies must be added to the controller system if more than two channels of I/O are required or if remote I/O is used.

1.3 Input/Output Section

The I/O section is the main interface to the user supplied devices (e.g. pushbuttons, limit switches, motor starters, solenoid valves, thumbwheels, numerical displays, and analog signals). The 584 Controller offers unsurpassed flexibility in allowing users to chose from a vast array of existing I/O devices. There are two I/O systems available for all portions of the 584's I/O section. A brief description of the I/O series follows with specific details on each I/O system provided in Section II of this manual.

The 200 Series I/O has maintained wide acceptance throughout the PC industry. Millions of I/O points are currently in use for a variety of applications. The 200 Series I/O modules provide either 16 input or 16 output circuits per module. Full error checking by redundant transmissions and echo checks assures maximum system integration from controller mainframe to the I/O modules. When cabinet installation is utilized, the 200 Series I/O requires a 14 inch deep NEMA enclosure.

As opposed to the 200 Series, the 500 Series I/O provide 4 input or 4 output circuits per module. Error checking via a CRC-16 code from the mainframe to remote interfaces, redundant transmissions, and echo checks are standard. In addition, sensors are provided on the local bus communications from the interface to the modules to detect hardware bus faults. The 500 Series I/O can be installed in an 8 inch deep NEMA enclosure.

Regardless of which I/O system is selected, several useful features are standard in both systems. The I/O modules are placed in heavy-duty housings to which user wiring is connected. The modules can be removed and replaced without interrupting field power or the controller's scan. This simplifies maintenance and minimizes down time. It is not required that the system be shut down to replace the modules. Each housing is also provided with a conduit for user wiring. The bare-wire clamp terminals on the I/O housing will accommodate one AWG No. 12 or two AWG No. 14 wires.

454

All circuits are fully isolated from the controller's logic. This enables the controller to withstand the severe voltage transients prevalent in industrial environments. Periodic maintenance is not required. Indicators are also provided which display the field power status and output fuse condition. I/O modules and field wiring can be configured without regard to voltage levels (i.e., 115 VAC vs. 24 VDC vs. 5 VTTL).

I/O can generally be grouped into two types, either discrete or register. Discrete I/O are signals that are simply ON or OFF. Examples of discrete I/O include pushbuttons, indicator lamps, motor starters, relay contacts, solenoid valves, limit switches, and relay coils. The MODICON 584 also has the capability to interface and control numerical devices, i.e. register I/O, such as thumbwheels, numerical displays, punched card readers, high speed counters, and rotational shaft encoders. These numerical I/O can be either BCD or binary (individually selected) and generally utilize more I/O capacity per device than discrete devices since more information is required than an ON or OFF status. The 584 Controller, as an option, can also be interfaced to ASCII devices such as CRT terminals, line printers, teletypes, and disk storage.

A summary of 584 Controller specifications is provided in Table I.

TABLE 1
BASIC 584 CONTROLLER SPECIFICATION

Power Requirements:		
Voltage:	115 VAC or 220 VAC ± 15% (Jumper Selectable)	
Frequency:	47-63 Hz	
Max. Load:	450 Volt Amps	
Peak Transient:	8 Amp at 115 VAC 4 Amp at 220 VAC	
Environmental:	Operation	Storage
Ambient Temperature:	0-60 C	-40-80 C
Humidity (non-Condensing)	0% to 95% RH	0% to 95%
Dimensions (W x H x D):		
Mainframe (incl. Power Supply)	19 in x 20 in x 16 in (485 mm x 510mm x 410mm)	
200 Series I/O Channel	20 in x 41 in x 13.5 in (510mm x 1045mm x 345 mm)	
500 Series I/O Channel	40 in x 32 in x 6 in (1020mm x 815mm x 155mm)	
Weight:		
Mainframe (incl. Power Supply)	60 lbs.	27Kg.
200 Series I/O (I Channel)	176 lbs.	79.2 Kg.
500 Series I/O (I Channel)	136 lbs.	62 Kg.

1.4 CRT Programming Panel

The main device used to program the 584 Controller is the P190 Programmer shown in Figure 4. This unit is a multifamily programmer capable of programming other MODICON PC's such as the 484 Controller. It incorporates a 9 inch CRT screen and a unique character generator designed specifically for relay ladder diagram displays. The tape drive incorporated into all P190 Programmers provides great flexibility in support of MODICON controllers. Three types of tapes are available for the MODICON 584 PC.

The UTILITY TAPE configures the user's system, as well as makes ladder listings of the user's logic. The PROGRAMMER TAPE allows the user to program the 584 in relay ladder logic. Lastly, the P190 TAPE LOADER TAPE allows the user to record the system's program onto a blank tape and reload it as required (see Figure 5).

FIGURE 5 INSERTING TAPE INTO P190

In addition to the tape drive, the P190 Programmer incorporates an ASCII keyboard and 31 fixed function control keys. These are packaged in a rugged case, which is easily moved to the work site or placed in a centralized location accommodating multiple remote controllers. The entire unit is ideally suited for an industrial environment; it is designed to operate in locations where electromagnetic noise, high temperature, humidity, and mechanical shock are prevalent.

The P190 Programmer connects directly to the 584 Controller providing a simple tool to program the processor via a ladder diagram. The "language" used to program the controller incorporates familiar relay symbology; there are no requirements to learn a new programming language. In addition, the CRT allows rapid and easy system checks and maintenance.

1.5 Other Peripherals

A number of standard MODICON support units besides the P190 Programmer are available for use with the 584 Controller. These units provide a wide variety of support functions, ensuring that complete system support is always available. As

455 over

previously discussed, the 584 features two peripheral ports, one on the front of the mainframe and the other on the bottom of the I/O Processing Board. Either port can be used to connect the PI90 or any other peripheral device; a maximum of two devices can be communicating to the 584 at one time.

Included in this group of support equipment are the T160 and T161 Telephone Interfaces for communication with MODICON'S SERVICE CENTER and a computer using MODICON's MODBUS protocol. These devices plug directly into either peripheral port without interrupting the controller's scan. Previous models of telephone interfaces such as the T151, T152, T154 and T158 can also be used with the 584 Controller by purchasing MODICON cable, MODICON NO. W192. These devices can connect to the mainframe through either peripheral port. For further details on the use of these peripheral units, see Appendix A.

Section II — Configuration

2.0 System Requirements

The 584 Controller and all its components are supplied in one enclosure referred to as the 584 mainframe (see Figure 1). Different memory sizes, memory types, logic capabilities, I/O capacity, and other options can be installed in this mainframe. Changes from one model to another can be accomplished by replacing internal printed circuit boards. Options that are available include:

- Memory Type and Size (16 bit words)

Core	CMOS
	4K
8K	8K
	12K
16K	16K
	24K
32K	32K

- Logic Capability
  - Basic Instruction Set
    - Relays, timers, counters, and arithmetics
  - Enhanced Instruction Set
    - Relays, timers, counters, arithmetics, move, matrix, skip, and ASCII
- Maximum Reference
  - 2048 discretes and 1920 registers (one word per programmed element)
  - 8192 discretes and 9999 registers (one and one half words per programmed element)

MODICON's 584 Controller has the capability to program or simulate the operation of relays, timers, and counters, as well as arithmetic operations (add, subtract, multiply, and divide). Additional functions such as MOVE, MATRIX, SKIP, and ASCII operations are optionally available.

To provide for efficient memory utilization in smaller systems and still accommodate capacity for large, I/O intensive systems, two memory versions are available. The small system is based on a l6 bit word size and has a maximum of 2048 discrete I/O points and 1920 registers. In the l6 bit word system, there is no appreciable difference between the CMOS and CORE memory; selection depends upon user requirements and cost.

The CMOS semi-conductor memory is equipped with lithium batteries. These batteries provide DC power to maintain user memory (programmed logic, register content, coil state, and internal system parameters) whenever external power fails or is removed by the user. It is recommended that the batteries be replaced approximately every l2 months. The user should note that the batteries are not designed to maintain system operation.

A "channel" of I/O represents 128 input points and 128 output points. Within a channel these separate inputs can not be traded for outputs, nor outputs for inputs. Thus, an I/O channel is a hardware definition for ease in wiring I/O signals to the controller. Thirty-two channels of I/O represent 4096 input points and 4096 output points; these I/O can be a mix of discrete (ON/OFF) and register (numerical) capabilities.

The specific I/O circuitry required to convert the various field voltages to signals that are compatible with the controller's processor is provided by the I/O module. Discrete I/O modules are either input or output; combining inputs and outputs on one module is NOT possible. Typical I/O modules are shown in Figure 6. These modules are installed in I/O housings (B545 or B546) for proper connection to field devices.

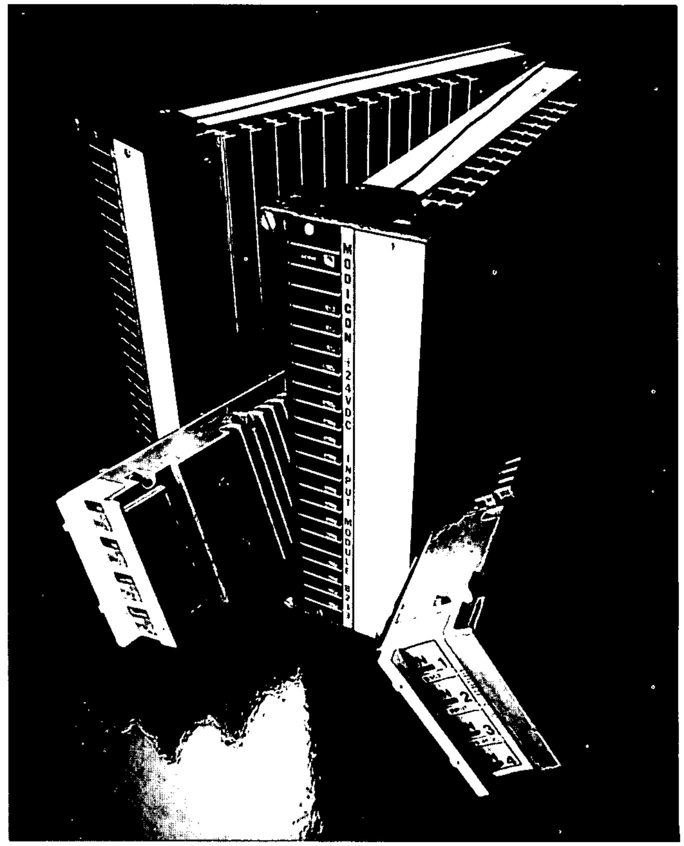


FIGURE 6 TYPICAL 200 AND 500 SERIES DISCRETE I/O MODULES

2.1 System Hardware Configuration

Figure 7 displays the installation dimensions and mounting hole locations for the 584 mainframe and associated units. For proper heat flow, all units should be placed vertically. This permits suitable ventilation via the heavy-duty housing fins. Keyhole type mounting holes are provided on the top and bottom of the mainframe to assist in mounting these units. Complete installation and checkout instructions are discussed in Section VI of this manual.

A full size mylar template displays the 584's mounting dimensions (Figure 7), as well as one channel of each type of I/O. Your MODICON representative will be glad to make this template available to you.

457 over

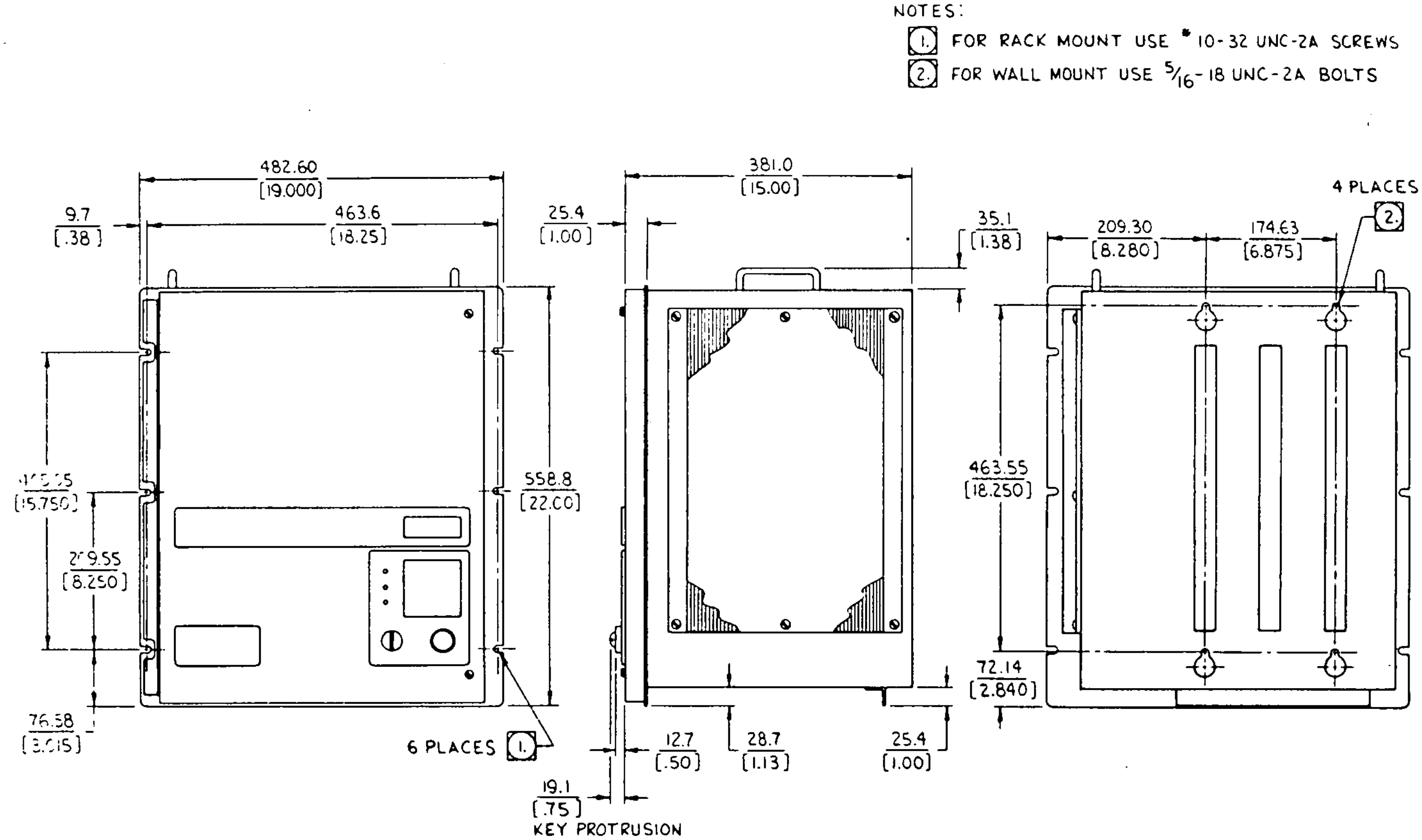


FIGURE 7 584 INSTALLATION DIMENSIONS

The remote I/O feature of MODICON's 584 Controller becomes a valuable asset for applications which fully utilize the controller's I/O capacity. Remote I/O allows the entire I/O channel or portions of a channel to be located thousands of feet from the mainframe. Communication with the mainframe is accomplished via a single coaxial cable or two twisted pairs. Remote I/O also allows the user to locate I/O adjacent to field devices. Each signal is wired a few feet from the I/O module, the signals are combined, and then sent to the mainframe by a single channel connection. The I/O section is further discussed in Section 2.3.I of this manual.

2.2 Mainframe Hardware Configuration

The mainframe is mounted directly into a backpanel, a vertical supporting member, or a 19 inch rack. The power supply is removable and contained in the right chamber of the mainframe. System operation will not be interrupted when the cover on the 584 mainframe is swung open to allow access to the internal components. System operation, however, will be shut down when the power supply or any other component is disconnected.

Proper power shutdown and power-up sequence in this event will not be performed.

NOTE: Controllers with CMOS memory will not be affected when the power supply stops operating as long as the batteries are functioning.

On the front of the mainframe (see Figure 8) is a six digit numerical display, a numerical keypad, three LED's, a keylock, and a MODBUS communications connector.

The numerical display is used to display maintenance information and, in conjunction with the keyboard, status of discretes, register content, and system data. Selective data can also be altered from the keyboard (see Appendix A for further details). Three LED's, when energized, indicate adequate battery voltage (Battery OK) if CMOS memory is used, proper output of the power supply (DC Power), and proper operation of the processor (RUN).

An indicator (Battery OK) on the front of the mainframe verifies proper battery voltage. When this indicator is extinguished, the batteries have sufficient power to support memory for 30 days. Lithium batteries are not rechargeable and have a shelf life of five years.

NOTE: AC power must be applied for the LED indicators, including Battery OK, to function. Loss of Battery OK LED indicates battery voltage is low. Batteries should be replaced at this time.

On the right side of the mainframe is a terminal board (see Figure 9) to which AC power is supplied. Table 2 summarizes the range of AC power over which the 584 Controller can operate. Voltage sensing circuitry is provided in the power supply to detect out-of-tolerance line voltages. If the AC power is not within proper specifications the processor will stop operating, forcing all outputs to the OFF condition and turning its RUN light OFF. Operation will be automatically restored when AC power is within tolerance. There will be a delay in restoring processor operation while the processor runs its power-up sequence which includes time to perform diagnostics after AC and DC power returns to operating voltages (approximately 3 seconds).

458

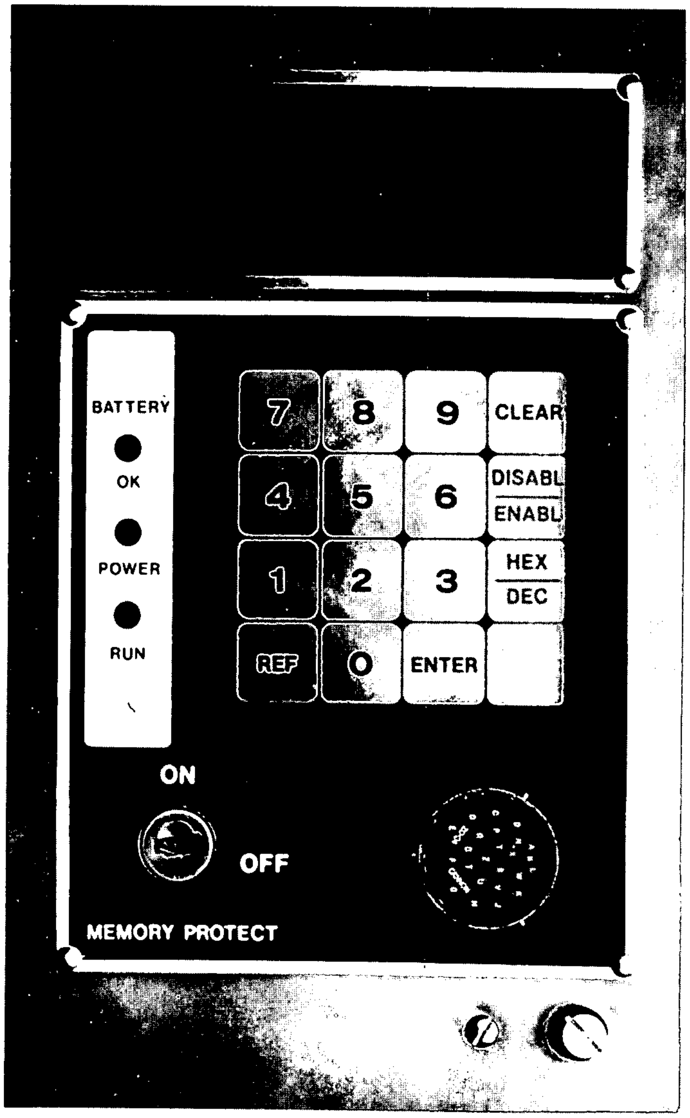


FIGURE 8 584 CONTROLS AND INDICATORS

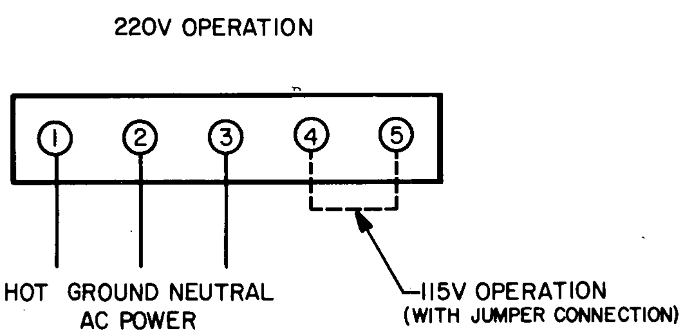
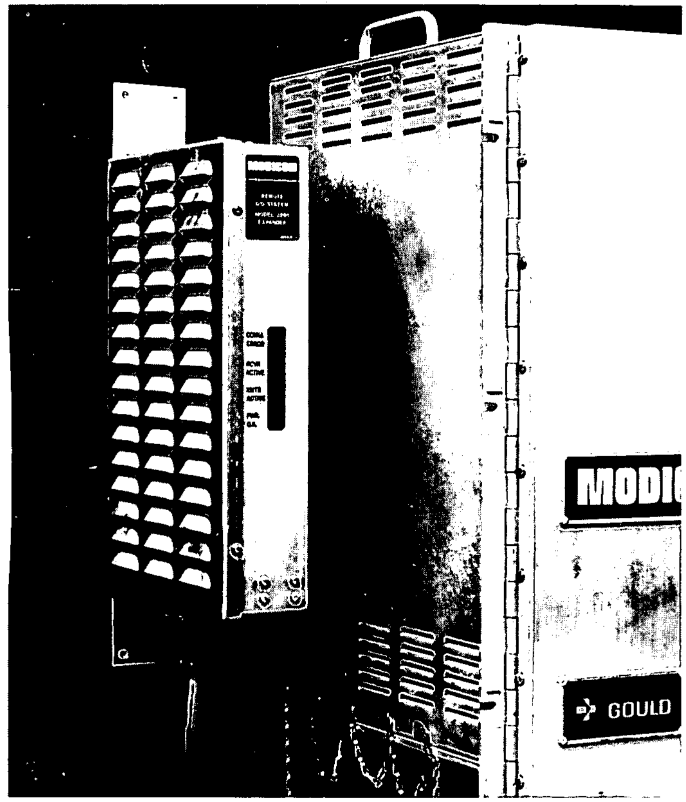


FIGURE 9 AC POWER CONNECTORS FOR 584 MAINFRAME



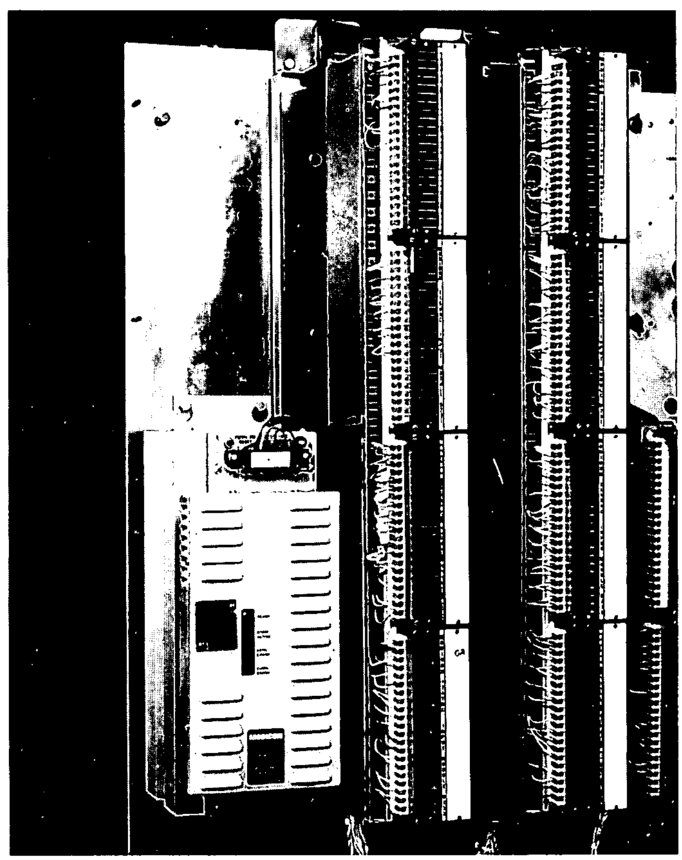


FIGURE 10 584 MAINFRAME MOUNTED WITH J200 I/O EXPANDER AND P451-X22 AUXILIARY POWER SUPPLY

459 over⇒

TABLE 2
SUMMARY OF REQUIRED AC POWER
FOR 584 MAINFRAME

Normal Voltage:		
	115 RMS ± 15% (100-130V RMS)	
	or	
	220V RMS ± 15% (187-253V RMS)	
	(Jumper Selectable)	
Transient Voltage:		
	Max. 10 Seconds:	115V RMS ± 30% (80-150V RMS)
		or
		220V RMS ± 30% (155-285V RMS)
	Max. 17 mSeconds:	115V RMS ± 100% (0-230V RMS)
		or
		220V RMS ± 100% (0-440V RMS)
	Line Spikes:	1000V max. (500 micro sec. duration 0.5% max. duty cycle)
Frequency:		47-63 HZ
Normal Load:		100 Volt-amps min.
		450 Volt-amps max.
		(depending upon I/O, memory, and peripheral devices connected).

2.3 Input/Output Section Hardware Configuration

The mainframe supports four local channels of I/O. When more than four channels are required, an I/O expander is used (see Figure 10). The model J200 Expander is capable of driving an additional twenty-eight channels of I/O, expanding the I/O capacity from four local channels to a total of thirty-two channels. All I/O connected to this expander can be distributed up to 15,000 feet from the mainframe. The twenty-eight channels of I/O can be placed in up to fourteen locations with a maximum of two complete channels at each location.

A channel of I/O contains up to 128 input points and 128 output points. These are separate maximum values; less I/O can be installed if required. There are no requirements to completely fill one channel before installing another. Further information regarding reference numbers appears in Section III and error checking in Section V of this manual.

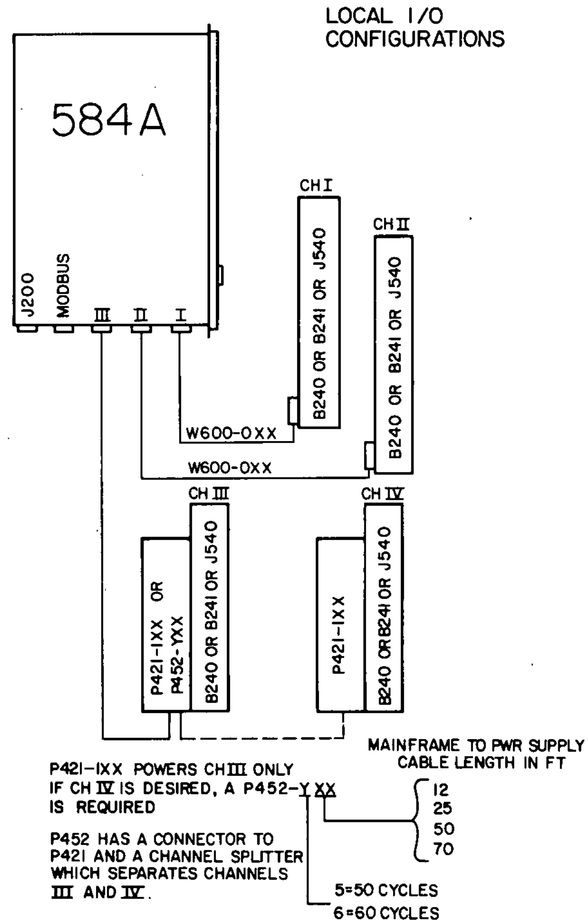


FIGURE 11A 584 LOCAL I/O CONFIGURATIONS

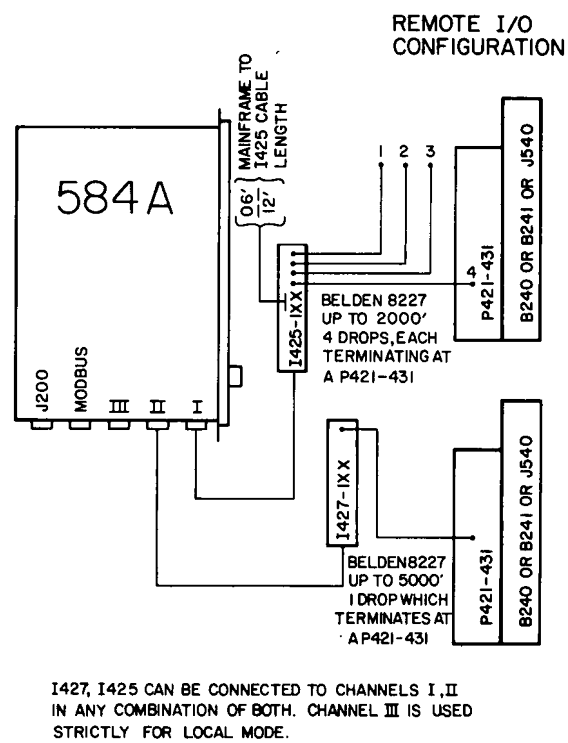


FIGURE 11B 584 REMOTE I/O CONFIGURATIONS

460

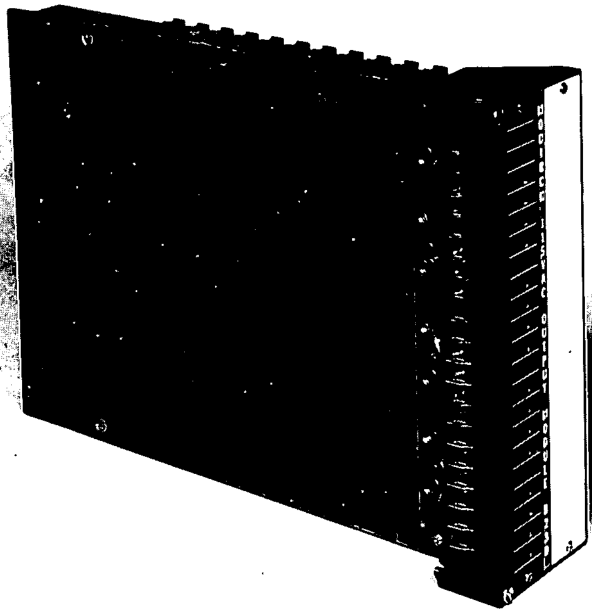


FIGURE 12 200 SERIES I/O MODULE

2.3.1 200 Series I/O System

Discrete I/O (ON/OFF) signals are provided by modules with 16 circuits (see Figure 12). These modules are either totally input or totally output; mixing I/O on one module is not possible. Numerical I/O data can be up to four BCD digits (maximum value 9999) or 12 bit binary for analog signals (maximum value 4095).

The 200 Series I/O modules are installed into I/O housings of which there are three varieties. The first, model B240, can accommodate up to 4 I/O modules. The second is a model B241 which can accommodate a maximum of 2 I/O modules. The B242 model prevents installation of non-intrinsic safe modules. Normally, four B240 housings are connected together to form a complete channel of I/O. (Up to 128 inputs and 128 outputs-eight input and eight output modules.) Since each channel is separately connected to the mainframe, only those I/O modules required in each channel need be installed.

Standard MODICON cables are used to connect each local I/O channel separately to the mainframe. These cables are heavy duty, multiple conductor, double-shielded cables available in the lengths shown in Table 3.

TABLE 3
MAINFRAME TO 200 I/O, CABLE OPTIONS

	Local	Via Auxiliary Power Supply	Via Remote Driver
Mainframe to Channel I & II	W600-003, 006	W602-012, 025, 050, 075	W604-006, 009, 012

NOTE The last three digits of the cable number represent the cable length in feet, except W600-003 which is 30" long.

Cables provided for connections to an auxiliary power supply are permanently attached. Cables for connection to the remote drivers must be ordered separately.

Figure 14 illustrates the size and mounting dimensions of a complete channel using full size housings, as well as the installation data for half size housings. The J540 adapter, which is used to interface to MODICON 500 Series discrete I/O, is also displayed.

When installing I/O housing, there occasionally exists a mechanical interference such as a mullun in an enclosure, that prevents normal insertion of I/O modules. To separate housings and thus avoid this type of interference, two special cables are available. A W608 (4 foot length) cable can be used to separate two housings, and a W609 (6 foot length) connects an auxiliary power supply to a housing or a J540 adapter. Unless these cables are used, I/O housings must be mounted adjacent to one another within a channel or remote I/O location.

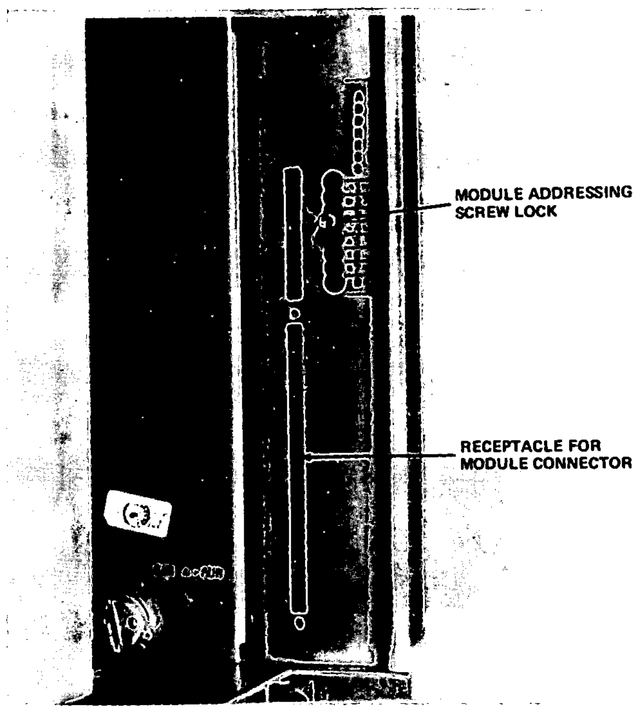


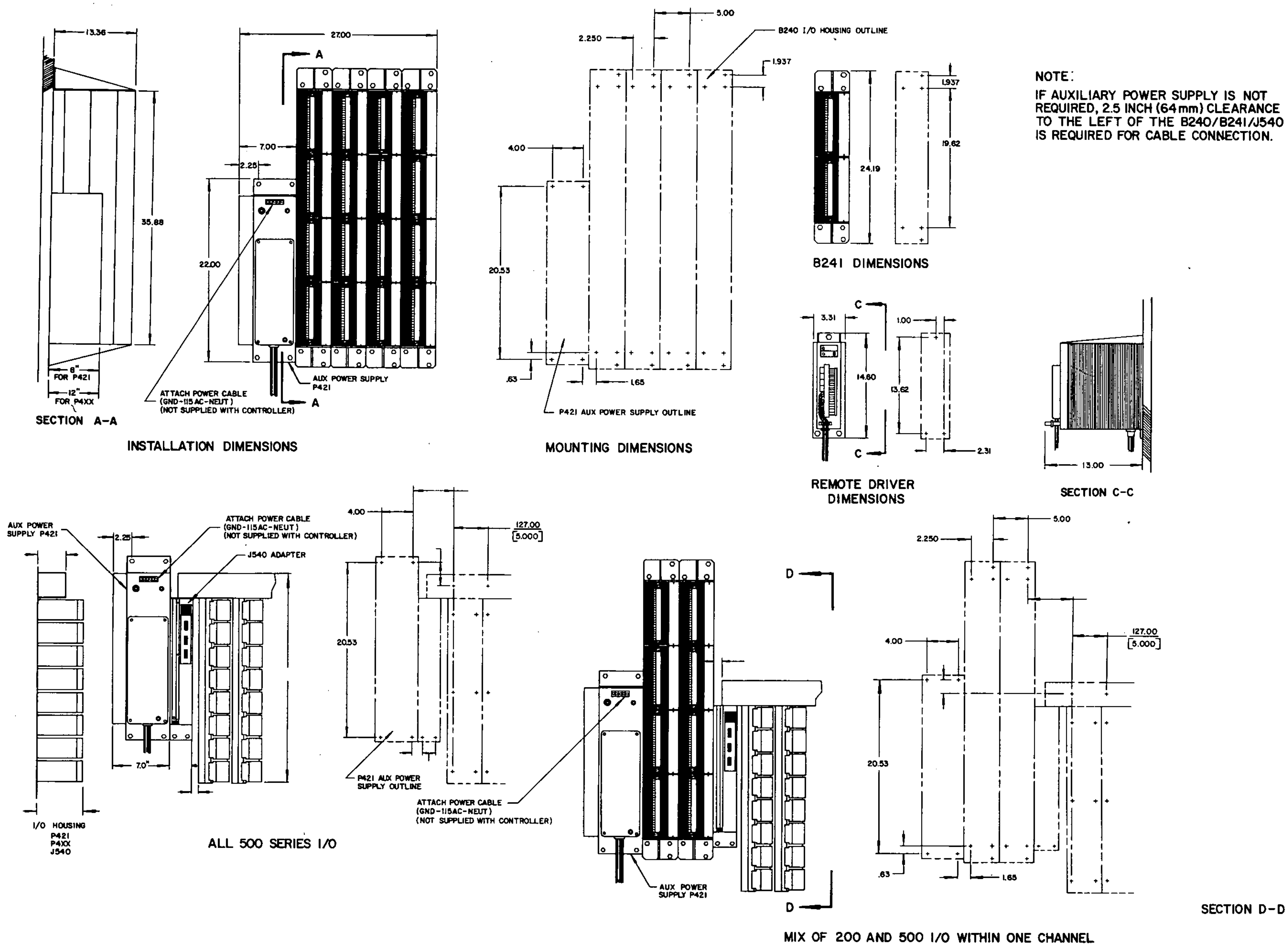
FIGURE 13 I/O HOUSING ADDRESS INDEX PIN

Within a channel of 200 Series I/O (four B240 housings or intermixing of B240's and B241's), I/O modules can be placed in any physical configuration desired. At the rear of each I/O housing are index pins, one per module (see Figure 13). Prior to installing the I/O modules, the index pins must be set (value one to eight) to indicate which of eight input or eight output modules are being placed in that location. The identification relative to input or output is automatically accomplished by the module itself. Thus, there can be two modules with the same index pin value, one input and one output.

Since the specific input or output identification is not established by physical placement of the module, but rather by the index pins, any convenient physical arrangement of I/O modules in a channel is possible. I/O can be placed with all inputs on the top and all outputs on the bottom, or all inputs on the left and all outputs on the right. Both options can be alternated if desired.

461 over→

NOTE:
IF AUXILIARY POWER SUPPLY IS NOT REQUIRED, 2.5 INCH (64 mm) CLEARANCE TO THE LEFT OF THE B240/B241/J540 IS REQUIRED FOR CABLE CONNECTION.



DIMENSIONS FOR J540 ADAPTER INSTALLATION

FIGURE 14 200 SERIES INSTALLATION DIMENSIONS

462

I/O modules that must utilize numerical values in lieu of discrete (ON/OFF) conditions can also represent more than one index pin location. If only these numerical modules are used, a complete channel can be a single four module housing. The flexibility of using index pins versus physical locations permits this savings. For example, if a numerical I/O module is placed in a channel and indexed to position one, adjacent physical locations in the housing can be used for any module type; the only caution is to limit the use of following index pin locations (values 2, 3, or 4). The following is a summary of I/O modules that use numerical values, the quantity of index pin locations they use, and the limits (if any) on pin location assignments.

INDEX PIN UTILIZATION

Module	Type	Quantity Input	Used Output	Must be Assigned to
B239	Dual Hi-Speed Counter (I/O)	2	2	Odd Pins
B243	Analog Input	4	0	1 or 5
B258	Analog Multiplexer	0	1	Any
B260	Analog Voltage	0	4	1 or 5
B262	Analog Current	0	4	1 or 5

Each I/O housing has on its lower left side a male printed-circuit connector and on its lower right a female receptacle. The male connector is normally retracted within the housing and is extended by rotating a cam, driven by a large screw head on the lower section of the backplane. Rotating this screw head 180 degrees clockwise extends the male connector; rotating it 180 degrees counterclockwise retracts the male connector. This connector is used to connect the housing to either a cable to the mainframe, an auxiliary power supply, or another housing. See Section VI for additional details on installing I/O housing.

When delivered, each housing has its male and female connectors, as well as its module backplane connectors covered by a protective tape. This tape must be removed prior to the connector's use. However, if the connector is not to be used (no module inserted or last housing in channel), the tape should be retained to ensure noise shielding and protect against entry of foreign matter.

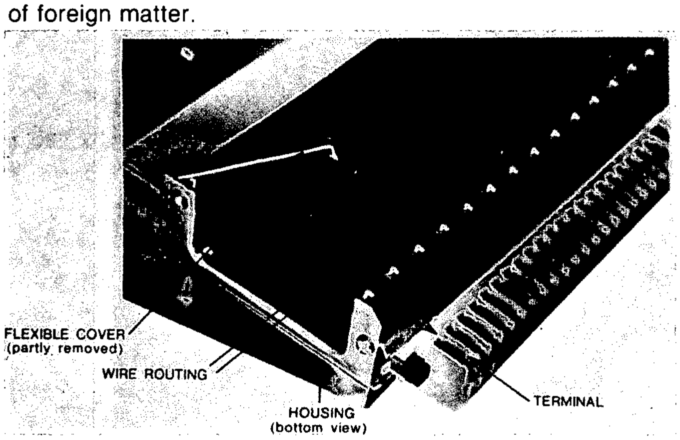


FIGURE 15 FIELD WIRING INSTALLATION

Field wiring (see Figure 15) can be installed on the I/O housing either before or after installation of the I/O modules. However, the address index pin must be positioned prior to installation. It is recommended that the field wiring also be fitted prior to module installation. Color-coded adhesive strips (Figure 16) are available to identify the 21 field-wiring terminals opposite each I/O module, terminal 1 (top) to 21 (bottom). These strips are color-coded to match the color of the installed module. This aids in preventing a module from being installed in a location not properly wired for that module type. These strips are available for each I/O module type and are installed by the user in accordance with his particular input/output configuration. The color codes are given in Table 4. Also provided with each I/O module is a white plastic plate so that the user can add his/her unique identification for each I/O circuit. This plate is reversible; both sides can be engraved.

TABLE 4
200 SERIES I/O PMS CODES AND UNITS OF LOAD/MODULE

Module	Type	PMS* Code	Color	Load (Per Module)
B230	115 VAC Output	199	Red	2 Units
B231	115 VAC Input	197	Pink	1 Unit
B232	24 Vdc Output	286	Dark Blue	2 Units
B233	24 Vdc Input	284	Light Blue	1 Unit
B234	220 Vac Output	151	Orange	2 Units
B235	220 Vac Input	149	Melon	1 Unit
B236	5V TTL Output	259	Violet	2 Units
B237	5V TTL Input	264	Light Purple	1 Unit
B238	24 Vdc Output, 2.5A	354	Green	2 Units
B239	Dual Hi-Speed Counter	515	Blue	3 Units
B243	Analog Input	109	Yellow	2 Units
B244	220 Vac Output, Isolated	465	Brown	2 Unit
B245	220 Vac Input, Isolated	465	Light Brown	1 Unit
B246	115 Vac Output, Isolated	233	Rhodamine Red	2 Units
B247	115 Vac Input, Isolated	231	Dark Pink	1 Unit
B248	10-60 Vdc Output	347	Green	2 Units
B258	Analog MUX	101	Yellow	2 Units
B260	Analog Output (Voltage)	380	Light Green	2 Units
B262	Analog Output (4-20 ma)	382	Green	2 Units
B266	Reed Relay Output	298	Silver Blue	2 Units
B270	48 Vac Outputs	207	Dark Red	2 Units
B271	48 Vac Inputs	204	Dark Rubine Red	1 Unit
B275	10-60 Vdc Input	314	Blue	1 Unit
J340	I/O Communicator	—	—	1 Unit
J342	I/O Comm. with Switchover	—	—	2 Units
J540	500 Series Adaptor	—	—	3 Units
J540/ B5XX	Adapter with one I/O Channel	—	—	13 Units
I425	Remote Driver	—	—	5 Units

*Pantone Matching System

463 over

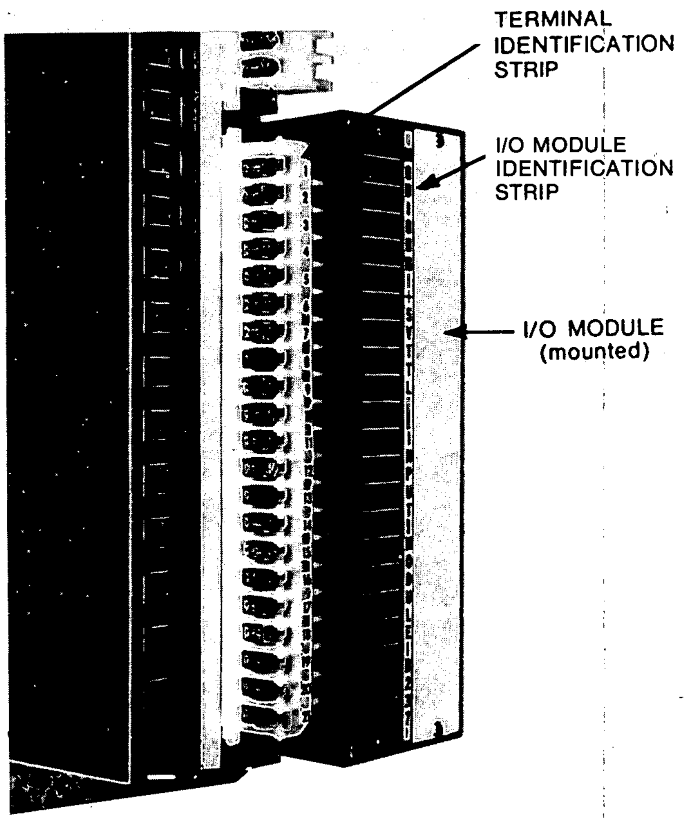


FIGURE 16 COLOR CODED ADHESIVE LABELS

The auxiliary power supply is not required for local I/O channels I and II, but is required for other channels and all remote locations. The power supply is available in either a 50Hz or 60Hz model (see Table 5). Both models can be operated from 115 AC or 220 AC by connecting jumpers as shown in Figure 17.

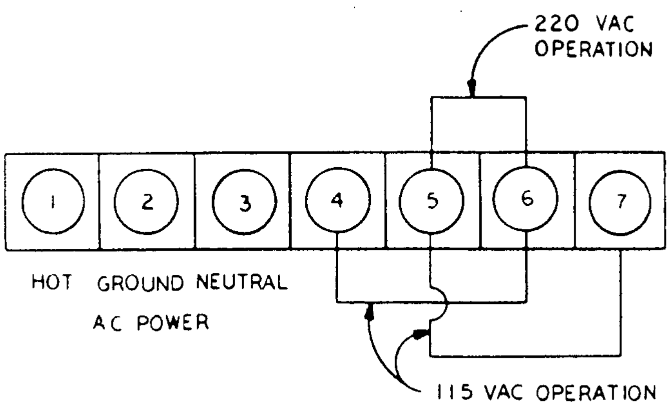


FIGURE 17 WIRING OF AUX POWER SUPPLY

TABLE 5
SUMMARY OF AC POWER FOR AUXILIARY POWER SUPPLY MODEL P421

Normal Voltage:	
Standard:	115V RMS ± 15% (100-130V RMS)
Optional:	220V RMS ± 15% (187-253V RMS)
(Jumper Selectable)	
Transient Voltage:	
Max. 10 Seconds:	115V RMS ± 30% (80-150V RMS) or 220V RMS ± 30% (155-285V RMS)
Max. 17 mSec	115V RMS ± 100% (0-230V RMS) or 220V RMS ± 100% (0-440V RMS)
Line Spike:	1000V Max. (500 micro. sec. duration, 0.5% Max. duty cycle.)
Frequency:	60 Hz ± 5 Percent (57-63 Hz) 50 Hz ± 5 Percent (47.5-52.5 Hz)
Normal Load:	10 Volt-amps (min.) 100 Volt-amps (max.) 4 amp peak on Transient (2 amps at 220 Vac)

NOTE: Exact load depends upon quantity of I/O modules installed.

SUMMARY OF AC POWER FOR AUXILIARY POWER SUPPLY MODEL P451

Normal Voltage:	
Standard:	115V RMS ± 15% (100-130V RMS)
Optional:	220V RMS ± 15% (187-253V RMS)
(Jumper Selectable)	
Transient Voltage:	
Max. 10 Seconds:	115V RMS ± 30% (80-150V RMS) or 220V RMS ± 30% (155-285V RMS)
Max. 17 mSec	115V RMS ± 100% (0-230V RMS) or 220V RMS ± 100% (0-440V RMS)
Line Spike:	1000V Max. (500 micro. sec. duration, 0.5% Max. duty cycle.)
Frequency:	
Standard:	60 Hz ± 5 Percent (57-63 Hz)
Optional:	50 Hz ± 5 Percent (47.5-52.5 Hz)
Normal Load:	300 Volt-amps (max. at 130 VAC) 2.7 amp peak at 130 Vac (2.0 amps at 253 Vac)

NOTE: Exact load depends upon quantity of I/O modules installed.

NOTE I/O modules are supplied with internal DC power from only one source; combining power from multiple sources is NOT possible.

Table 4 summarizes the load each 200 Series I/O device places on its power source. The main power supply has 54 units of I/O power available, and each auxiliary power supply can drive 27 units of I/O load. More than four I/O housings can be utilized on any channel as long as the I/O modules do not overload the power supply nor are there more than eight unique input or output addresses.

NOTE: When referring to Table 4, a 200 Series load can be calculated as follows:

300 mA X (No. of unit loads) = total current load

When referring to Table 7, 500 Series output loads can be determined by:

.23 X 300 mA X (No. of unit loads) = total current load

500 Series input loads are equivalent to:

.06 X 300 mA X (No. of unit loads) = total current load

Table 7 displays 0 mA since input load is minimal.

A complete I/O channel consisting of eight input modules (one unit of load per module) and eight output modules (two units per module) represents a total load of 24 units. The main power supply has approximately twice the capacity of an auxiliary power supply, providing it the capability to operate two complete channels, Channel I and Channel II. An auxiliary power supply must be used on each subsequent I/O channel.

NOTE: The main power supply can deliver a maximum 27 units of load to both Channels I and II, for a total of 54 units of load.

Special AC I/O modules are available which allow individual power sources on each circuit.

These I/O modules are termed "isolated" since separate pairs of field terminals are provided for each circuit; they incorporate no special isolation relative to electrical noise. See Appendix B for details.

Finally, the 200 series I/O also offers intrinsic safe input modules. These modules require special I/O housing (model B242) which prevents installation of non-intrinsic safe modules. Otherwise, these housings are identical to B240 and B241 housings.

Occasionally data must be transferred from one 584 mainframe to another. This can be accomplished by wiring an output module on one controller to an input module on another. However, for additional capacity, a mainframe to mainframe communicator (model J340) can connect an entire channel between controllers. This communicator provides the equivalence of eight modules of data in each direction, without installation or wiring of additional I/O modules. See Appendix A for details.

NOTE: The J340 Communicator is also compatible with 184 and 384 Controllers. This allows interconnection of a 584 to either a 184 or 384. A maximum of two J340 communicators can be used on one 584 controller.

Each I/O module incorporates an "active" light that indicates mainframe to I/O module communication is occurring. This indicator is extremely valuable in troubleshooting the I/O system. Additionally, each circuit (both inputs and outputs) has an indicator that displays the status of field terminal voltage. These indicators are used to troubleshoot the interface between the controller and external field devices.

NOTE: Input circuit status indicators will operate even if mainframe power and communications are not available. They depend only upon field voltage to obtain maximum reliability. AC output modules also have indicators which are energized if an output fuse should fail.

Fuses used on modules which accommodate field replacement are listed in Table 6. To replace a fuse, remove the module from its housing. Access to the fuse is provided by an opening (approximately I inch X 5 inches) on the terminal side of the module. All fuses are oriented in accordance with the output terminals such that the top fuse is for the No. 1 output and the bottom fuse is for the No. 16 output; except for the B238, whose top fuse is for the common indicator supply, and the B244 and B246, whose orientation is as shown in Figure 18.

TABLE 6
200 SERIES FUSE REQUIREMENTS

Module	Standard Size Pico Fuse	Littlefuse Part No. or Equivalent	Quantity per Module
B230	5 amps	275-005	16
B232	7 amps	275-007	1
B234	5 amps	275-005	16
B236	2 amps	275-002	1
B238*	3 amps	275-003	17
B243	1/4 amp	275-250	8
B244*	7 amps	275-007	8
	1/4 amp	275-250	1
B246*	7 amps	275-007	8
	1/4 amp	275-250	1
B248	3 amps	275-003	16
B258	1/2 amp	276-500	1
B266	3 amps	212-003	8
B270	5 amps	275-005	16

NOTE: Those modules indicated by an asterisk (*) are provided with one fuse for each output circuit plus one fuse for separate indicator lamp supply.

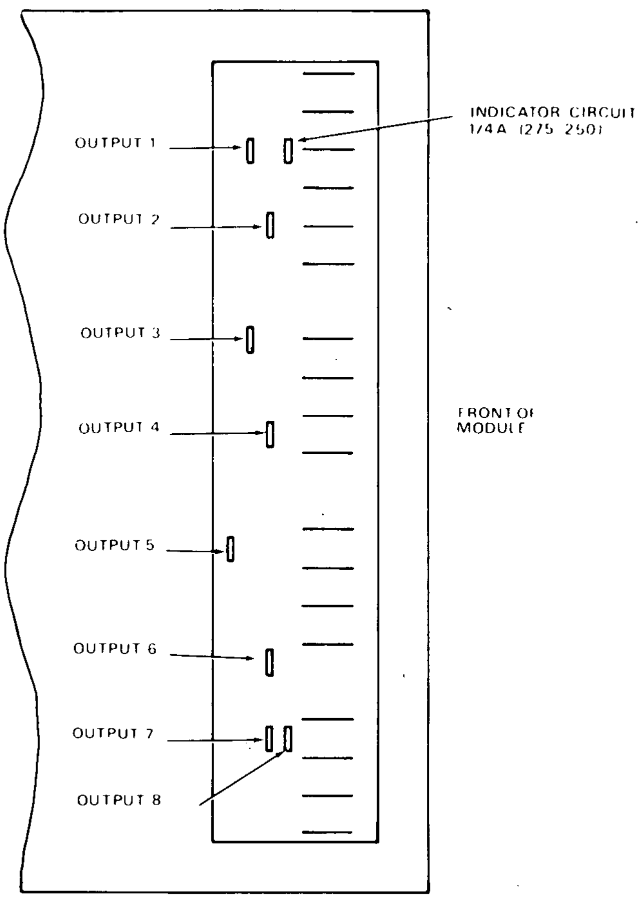


FIGURE 18 B244 and B246 FUSE LOCATION

465 over

2.3.2 500 Series I/O System

The 500 Series discrete I/O, i.e. ON/OFF signals, provides 4 circuits per module (see Figure 19). These modules are either totally input or output; in other words, a mix of input/output on one module is not possible.

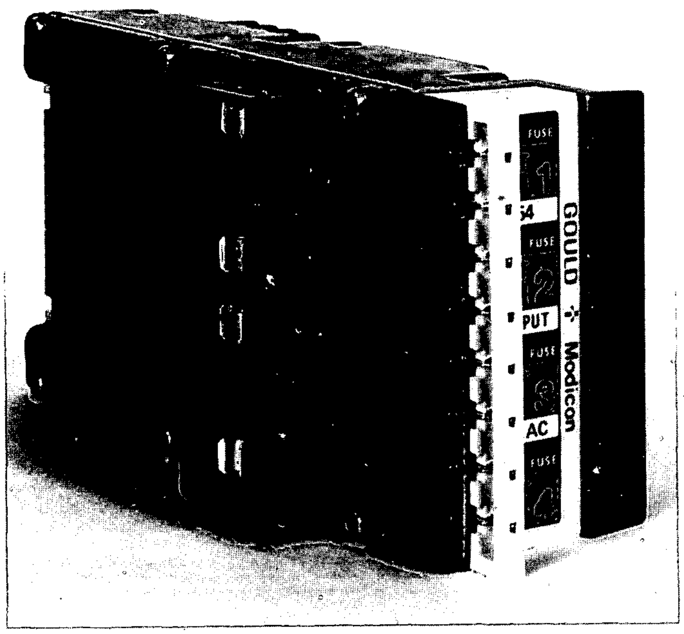


FIGURE 19 500 SERIES I/O MODULE

NOTE: The 584 Controller can be programmed to interface discrete I/O modules with single numerical devices.

The 500 Series I/O modules are installed into I/O housings (Model B545 or B546), each housing capable of receiving up to eight I/O modules. Normally, housings are connected to form a complete channel of I/O (up to 128 inputs and 128 outputs), which allows up to 32 input modules and 32 output modules to be installed in each channel. Since each channel is separately connected to the mainframe, only those I/O modules required in each channel need be installed.

The 500 Series I/O can be utilized on the four local channels with the standard 200 Series cables and power supplies and the J540 Adapter.

NOTE: The optional I/O Housing (model B546) accommodates a maximum of four modules and is approximately half as tall as the B545 Housing.

Since 128 inputs can be supplied on 32 input modules (4 inputs per module) and the same number of outputs also require 32 output modules, an entire channel requires 64 I/O modules. Thus, a complete channel of 500 Series I/O can be installed in eight B545 Housings. These housings are connected across their tops via a metallic duct. Contained within the duct is a bus cable type W510. The bus cable is connected to the edge connector at the top of each housing as shown in Figure 20.

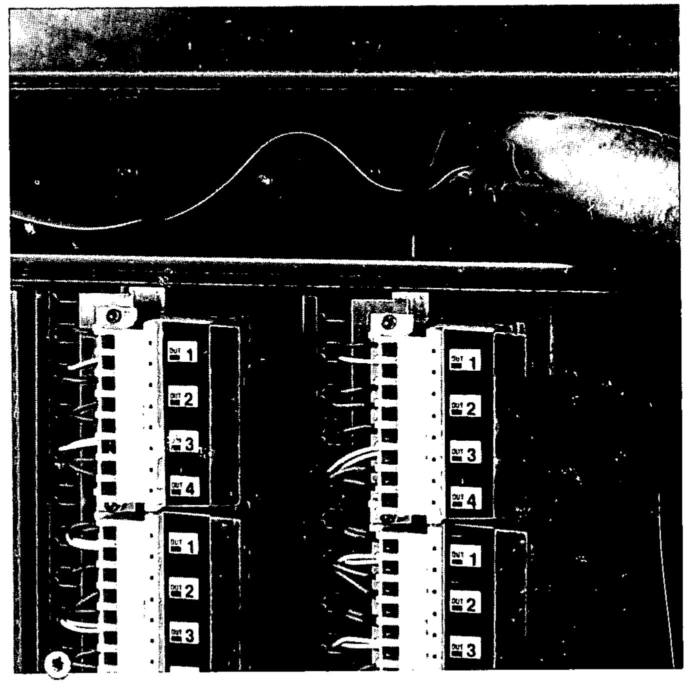


FIGURE 20 INSTALLATION OF BUS CABLE

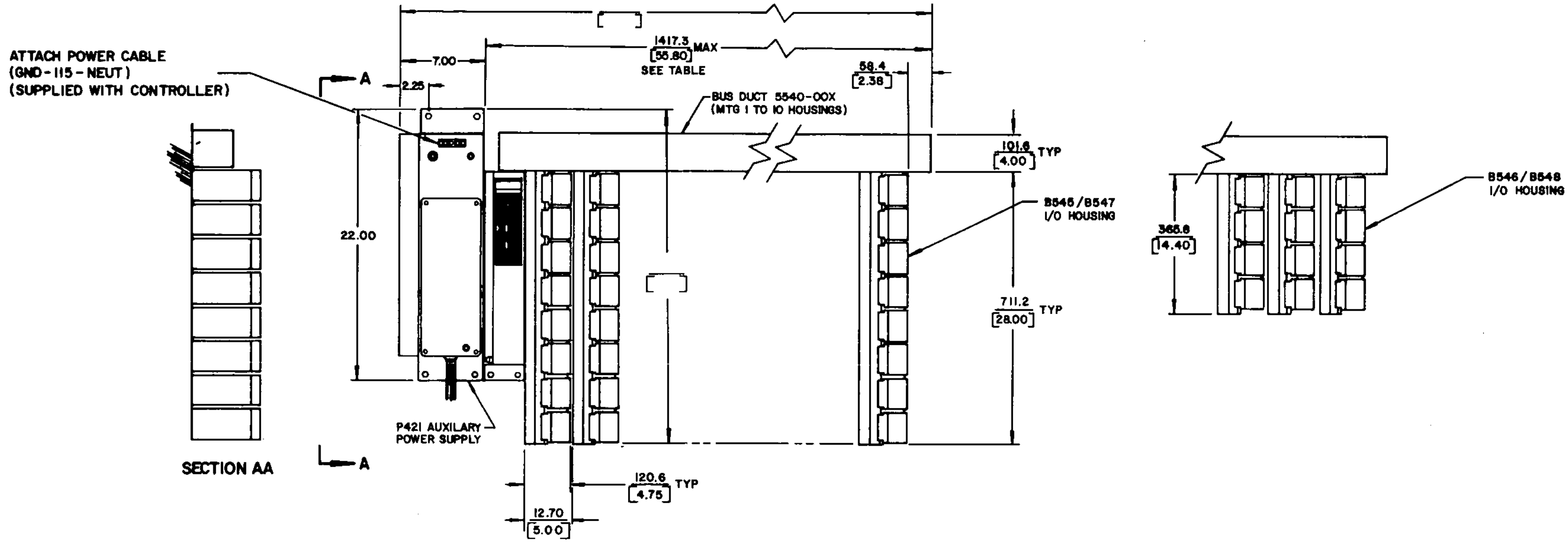
Figure 21 illustrates the size and mounting dimensions of a complete I/O channel using full size housings, as well as installation data for half size housings. The size and mounting information for the J540 Adapter is also shown. For exact determination of power supply load when using 500 Series I/O, see Table 7. Auxiliary power is available in either 50 Hz or 60 Hz (see Table 5). The models can be operated from 115 Vac or 220 Vac by connecting jumpers as shown in Figure 17.

Addressing is accomplished at each housing, 32 I/O points per housing within each channel. At the top of each housing is a set of four switches (see Figure 22) to address that housing; one switch is closed (moved towards field terminals) to select address. Any housing can have any address from one to four; the address does NOT depend upon physical position, but by the position of these switches. Within a housing, the eight modules (four input or output points per module) can be of any type (input or output). However, since there are a maximum of eight housings and only four address positions, two housings can have the same address. Any two housings with the same address must have an I/O configuration which is the exact opposite of each other; thus, the top module in one must be an input and the other an output. The same is true of every module position in the housing pair.

NOTE: If two output modules are in the same position of identically addressed housings, both modules will have the same status. When two input modules have the same address, the inputs will be the logical OR of these modules (OFF only if both OFF).

A channel with four I/O housings can be addressed 1 to 4 with I/O modules inserted in any order. There is no limitation relative to voltage levels. Beyond four housings (128 I/O points), each housing must contain the exact opposite (input versus output) of the previous similarly addressed housing. For example, if housing number 2 contains the following type modules (top to bottom, I = input, 0 = output): I,I,0,0,0,I,0,0, the other housing

466



INSTALLATION DIMENSIONS

10 HOUSINGS	55.80 MAX	1417.3 mm
9 HOUSINGS		
8 HOUSINGS	45.80 MAX	1163.3 mm
7 HOUSINGS		
6 HOUSINGS	35.80 MAX	909.3 mm
5 HOUSINGS		
4 HOUSINGS	25.80 MAX	655.3 mm
3 HOUSINGS	20.80 MAX	528.3 mm
2 HOUSINGS	15.80 MAX	401.3 mm

BUS DUCT LENGTH

TABLE I

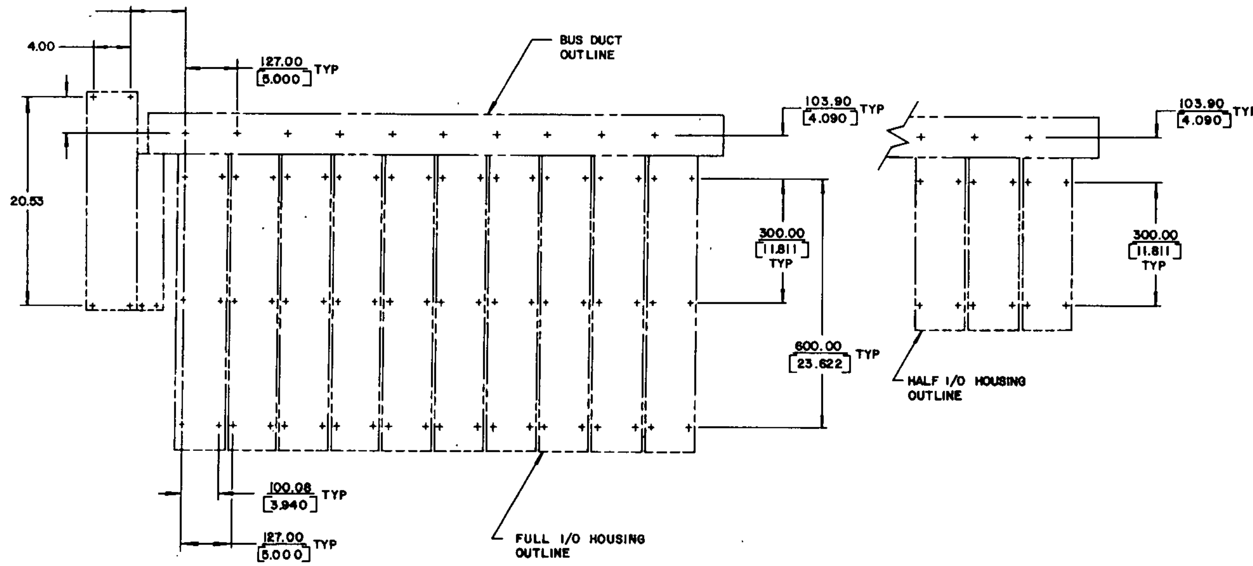


FIGURE 21 500 SERIES MOUNTING DIMENSIONS

467 over →

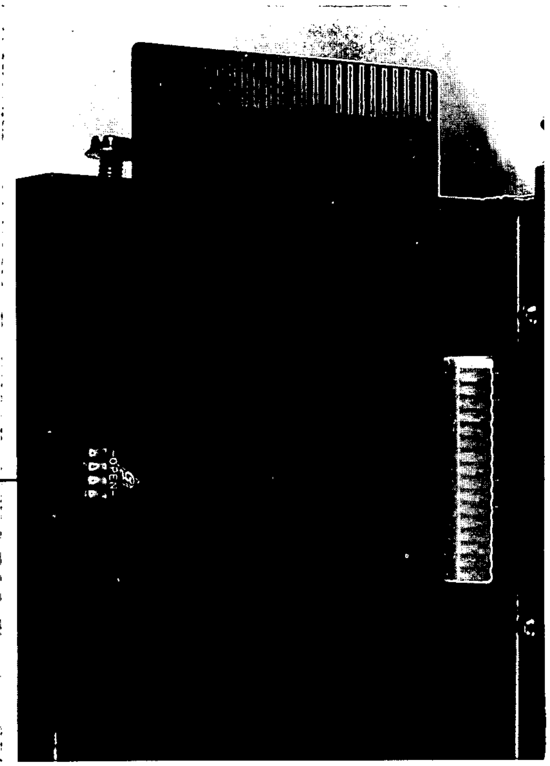


FIGURE 22 500 SERIES I/O HOUSING ADDRESS SWITCH

addressed to number 2 must contain: 0,0,I,I,I,0,I,I. At a maximum, housings are addressed in pairs; no more than two housings can be addressed to the same value.

In addition to the STRIP SELECT set of switches, the half-size I/O housing (B546) contains two additional sets labeled UPPER BYTE SELECT and LOWER BYTE SELECT. These switches select, respectively, the upper and lower references for each pair of modules attached to the housing. Four dip switches comprise each set.

To establish the modules' upper and lower references, one dip switch is positioned towards the field wiring terminals. For example, the UPPER BYTE SELECT dip switch number I selects modules one and two, the next dip switch selects modules three and four, etc., concluding with dip switch number 4 which selects modules seven and eight. The LOWER BYTE SELECT dip switches function in the same manner.

Field wiring (see Figure 23) can be installed on the I/O housings either before or after the I/O modules are installed. However, for user convenience, it is recommended that the field wiring be installed prior to the I/O modules. The I/O housing address can be re-adjusted at any time the I/O modules are inserted or removed. To set housing address, the top I/O module must be removed.

Color coded adhesive strips (see Figure 24) are available to identify the field wiring terminals opposite each I/O module, terminal 1 (top) to 8 (bottom). These strips are color coded to match the color of the module to be installed; this aids in preventing a module from being installed in a location not properly wired for the module type. Strips are available for each I/O module type and are installed by the user in accordance with his particular input/output configuration. The color codes are given in Table 7.

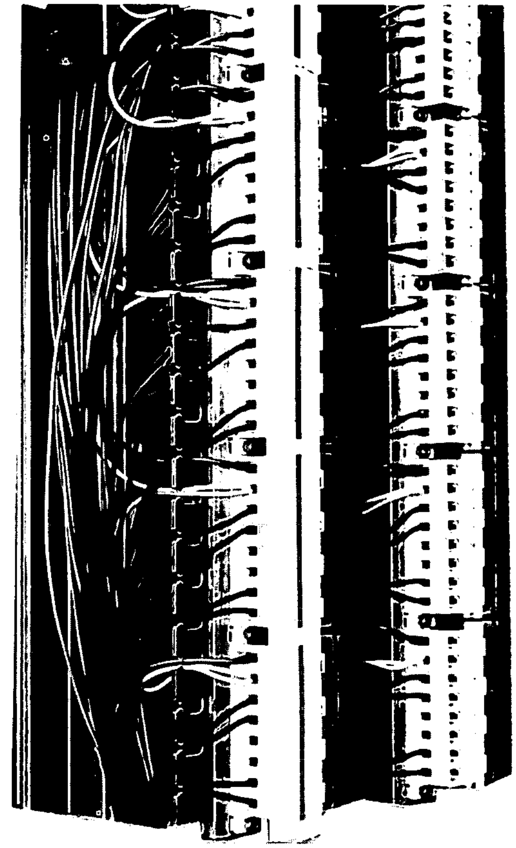


FIGURE 23 FIELD WIRING INSTALLATION

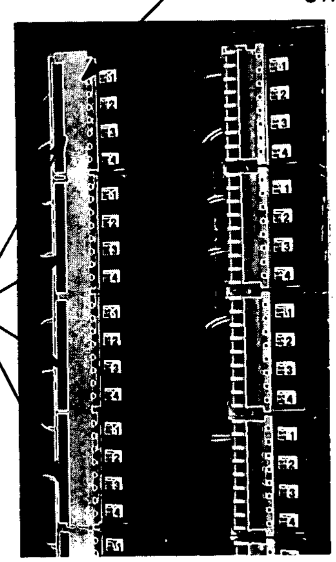


FIGURE 24 COLOR CODED ADHESIVE LABELS

468

TABLE 7
500 SERIES I/O MODULE PMS CODES AND UNITS OF LOAD/MODULE

Module	Type	PMS* Code	Color	Load (Per Module)
B550	115 Vac Output	199	Red	1
B551	115 Vac Input	197	Pink	0
B552	DC True High Output	286	Dark Blue	1
B553	DC True High Input	284	Light Blue	0
B554	220 Vac Output	151	Orange	1
B555	220 Vac Input	149	Melon	0
B556	5V TTL Output	259	Violet	1
B557	5V TTL Input	264	Light Purple	0
B558	DC True Low Output	314	Turquoise	1
B559	DC True Low Input	311	Blue	0
B560	120 Vdc Output	307	Blue	1.5
B561	120 Vdc Input	305	Blue	0
B562	DC Clamped Output			1
B564	24/48 Vac Output			1
B565	24 Vac Input			0

*Pantone Matching System

Portions of the I/O section can be distributed from the mainframe at distances up to 15,000 feet. The 500 series I/O used in this manner will require an auxiliary power supply and a J540 adapter. Communications over the remote I/O system are protected by a CRC-16 error checking code and require a positive response from the receiving device.

NOTE: Remote I/O can make major savings on installation costs and simplify maintenance. However, it does not affect scan time nor does it add more I/O than the controller allows per channel.

All I/O modules require DC power to operate their internal circuitry for communications to the mainframe. This power is supplied either from the mainframe power supply or from an auxiliary power supply.

NOTE: I/O modules are supplied with internal DC power from only one source; combining power from multiple sources is NOT possible.

Table 7 summarizes the load each discrete I/O device places upon its power source. Each auxiliary power supply can drive 27 units of load.

NOTE: Units of load are convenient measurements of load designed for easy computations. 500 Series I/O units of load are not identical to those of 200 Series I/O, although they perform a similar purpose.

More than eight discrete I/O housings can be utilized on any channel as long as the I/O modules do not overload the power supply and there are NOT more than four unique B545 I/O housing addresses.

The 500 series I/O also offers intrinsic safe input modules. Intrinsic safe modules are enhanced by special circuitry design which allows the modules to function in the most hazardous industrial conditions. These modules require special I/O housings (model B543/B544) which prevent installation of non-intrinsic safe modules. Otherwise, these housings are identical to B545 and B546 housings. Each circuit (both inputs and outputs) also has an indicator that reflects the status of the field terminal voltage. These indicators are used to troubleshoot the interfaces between the controller and external field devices.

NOTE: Most input circuit status indicators will operate even if mainframe power and communications are not available.

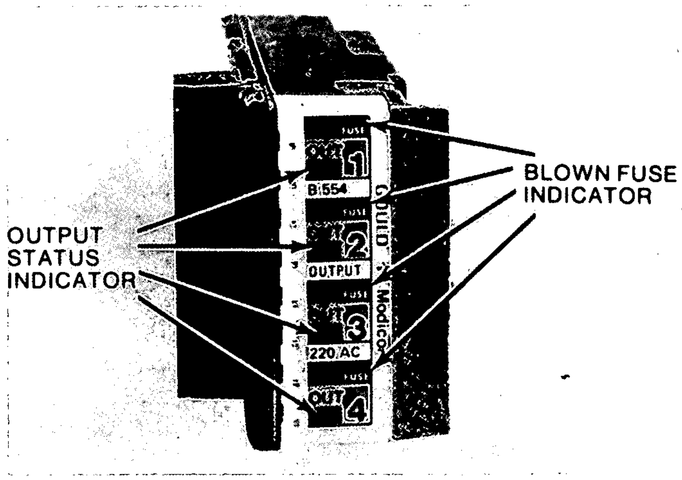


FIGURE 25 500 SERIES I/O INDICATOR LIGHTS

All output modules (AC or DC) except B556 (5V TTL) are fused with field replaceable fuses and incorporate a blown fuse indicator. The blown fuse indicator is viewed from the front of the module as shown in Figure 25; one blown fuse indicator is provided per circuit. To replace fuses, the module must be removed from the I/O housing.

NOTE: When a module is removed, all four circuits (input or output) will be disconnected.

All fuses are 3AG, normal blow. The AC output modules utilize a 5 amp size, and the DC output modules utilize a 3 amp size. Once the module is removed, fuses are easily removed from the module's left side (see Figure 19.)

469 over→

2.3.3 Remote I/O - I425

Portions of the local I/O can be remoted from the mainframe on a channel by channel basis, with distances of up to 2000 feet. Channels I or II can be remoted by installing a model I425 Driver on the channel at the mainframe; the driver can be located a maximum of 12 feet from the mainframe. Channels beyond the four local channels are designed for remote I/O with the J200 remote I/O driver. These channels can be distributed up to a total of 15,000 cable feet from the mainframe. Regardless of which method is used to remote a channel, each location requires an Auxiliary Power Supply equipped with a remote Interface (model P421-431 with I425 Driver or P451-X22 with J200 Expander) and can drive up to one full channel (8 input modules and 8 output modules).

Figure 26 displays a wiring diagram for connecting the I425 to a P421-431.

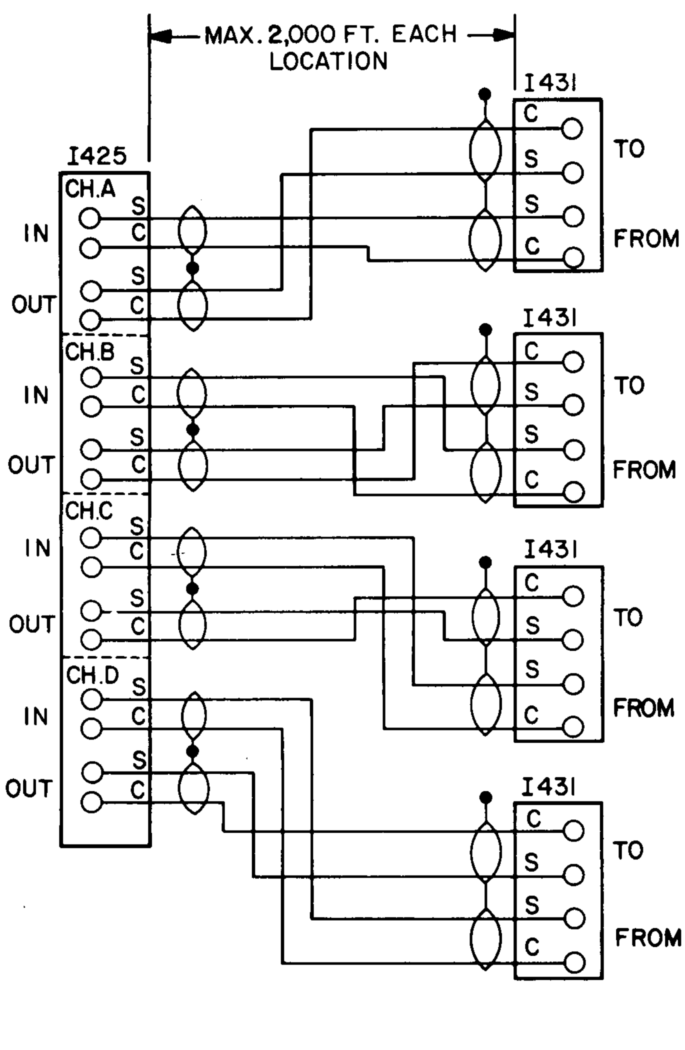


FIGURE 26 TYPICAL REMOTE I/O WIRING USING THE I425

2.3.4 Remote I/O - J200

The J200 Expander can support up to fourteen I/O locations, each with up to two I/O channels. A single RG-6/U, (CAC-6) or CATV cable run is used to connect all I/O locations (up to 22) to the expander. This cable is configured as a multi-drop connection wherein a single main cable is installed and taps used to connect each I/O location to the main cable. Maximum length of any tap is 100 feet (30 meters); total cable length should not exceed 5,000 feet (1.8 Km) for RG-6/u or 15,000 feet (4.5 km) for CATV.

The configuration of J200 and P451's will normally require custom configurations for each installation. To aid in laying out these installations, the following guidelines should be used.

1.) The maximum allowable dB loss between the J200 and any P451 is 35dB.
2.) The maximum allowable cable run is 15,000 feet (4570M) even if the dB loss is less than 35dB.
3.) The dB loss for the cabling is as follows:
CATV = 0.8dB/1000 ft. (305M)
RG-6/u = 7.0 dB/1000 ft. (305M)
4.) The dB loss for the connectors is as follows:

Tap (DT-10)	1dB through, 12dB down drop
Splitter (DS-2H)	3dB from center to both sides

Figure 28 shows a typical interconnection diagram for a J200 installation. The figure includes all the cables, connectors and terminators that any installation in any configuration could implement. Table 8 lists recommended part numbers and suppliers for cables and connections used in Figure 28.

To aid the user in understanding the information provided in Figure 28 and Table 8, the following describes sample dB loss calculation for two drops.

A splitter is used at the J200 to create a branch in the system and allows the cable to be laid out in two directions. The purpose of this is to minimize the dB loss from the J200 to any P451 drop. This will create an immediate 3dB loss in both directions out of the splitter. If the distance from the J200 to the splitter is 10 feet and from the splitter to the first tap on the left is 2000 feet, a sample dB loss calculation would look as follows:

10 ft.	RG6/U cable	0.07	dB
1	Splitter	3.00	dB
2000 ft.	CATV cable	1.60	dB
1	TAP	12.00	dB
100 ft.	RG6/U cable	0.70	dB
	TOTAL	17.37	dB

If the 2nd tap to the left is located 3000 feet from the first tap, the calculation would look like:

10 ft.	RG6/U cable	0.07	dB
1	Splitter	3.00	dB
2000 ft.	CATV	1.60	dB
1	TAP	1.00	dB
3000 ft.	CATV	2.40	dB
1	TAP	12.00	dB
100 ft.	RG6/U cable	0.70	dB
	TOTAL	20.77	dB

470

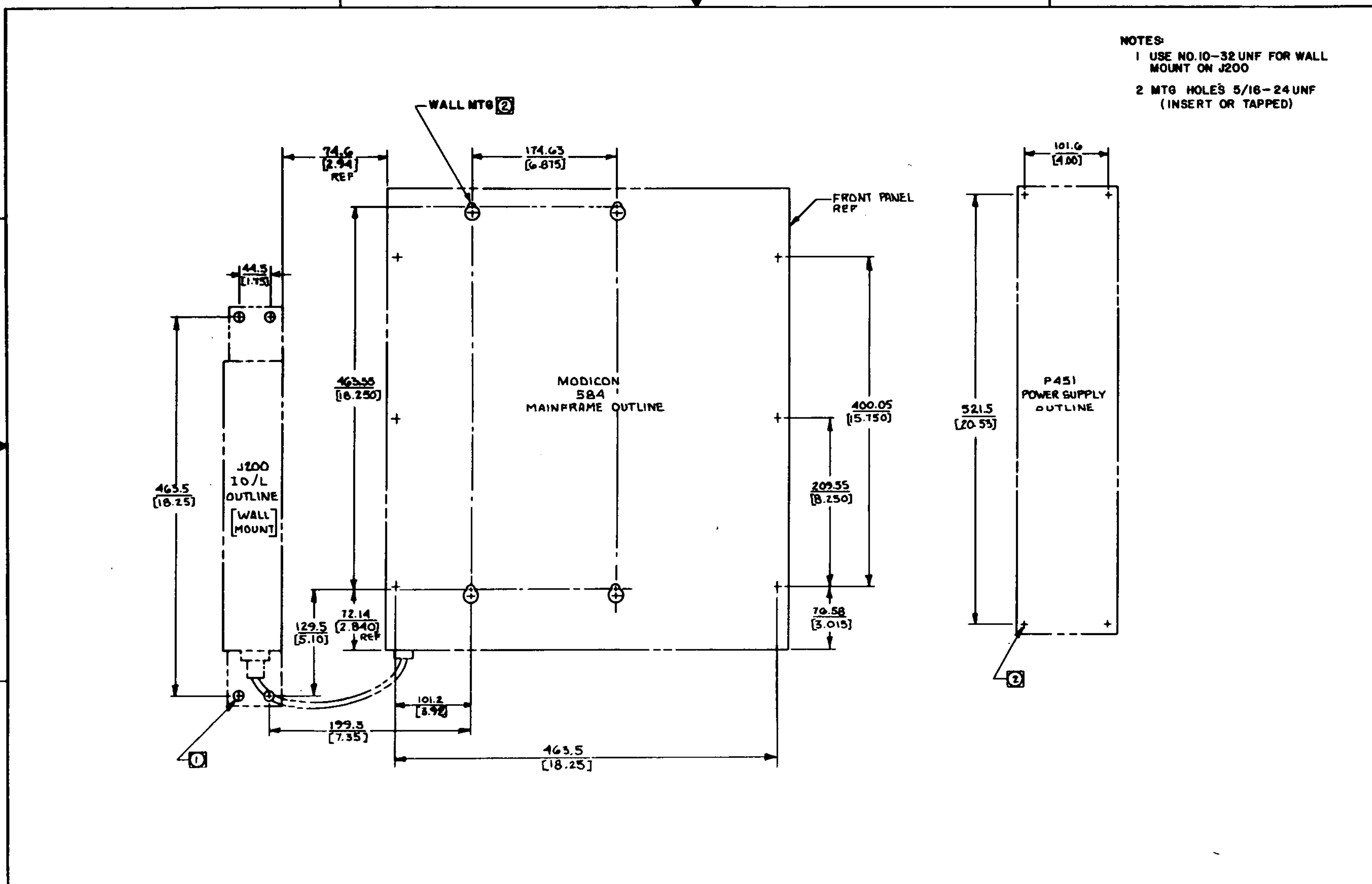


FIGURE 27(A) J200 AND P451-X22 MOUNTING DIMENSIONS

471 over →

The left and right side of the drawing show the 2 options for termination of the cable. The left side terminates at a P451 -X22 power supply which would have I/O hooked up to it. The right side has a terminator to end the cable run. The advantage of this latter approach is the ease with which the system can be expanded to add additional drops. (Run the cable, hook up all the additional drops, and then disconnect the terminator and attach the additional cable run.)

The configuration drawing uses Taps to make all the drops, which adds 12dB in losses to each drop. As the cable run becomes longer, this may create a situation which will exceed the 35dB loss and prevent installation of a drop at this site. To avoid this problem, splitters may be used in place of a tap. This has the advantage of giving only a 3dB loss down the drop, but a disadvantage in that it creates a 3dB loss through the splitter to the next drop. Therefore, the use of splitters will typically be on the drops furthest from the mainframe.

TABLE 8
PARTS LIST FOR CONNECTORS,
TAPS AND TERMINATORS FOR FIGURE 28

1. **CATV CABLE**
   A) Comm-Scope Co.
      Parameter III — Model No. P-3-75-500-JCA
   B) Times Co.
      Lumifoam III — Model No. JT-4500-J (Jacket required for insulation. See caution.)
2. **CAC-6 CABLE**
   Belden — Model No. 9283
3. **CATV CABLE TO MALE F CONNECTOR**
   Gilbert Engineering — Model No. GRS-500-AFM-DU03
4. **CABLE TERMINATOR - 75 Ohm**
   Taco/Jerrold — Model No. TR-75F
5. **TAP**
   Extronix — Model No. OT-10
6. **SPLITTER**
   Extronix — Model No. DS-2M
7. **TYPE F MALE CONNECTOR**
   Taco/Jerrold — Model No. F56

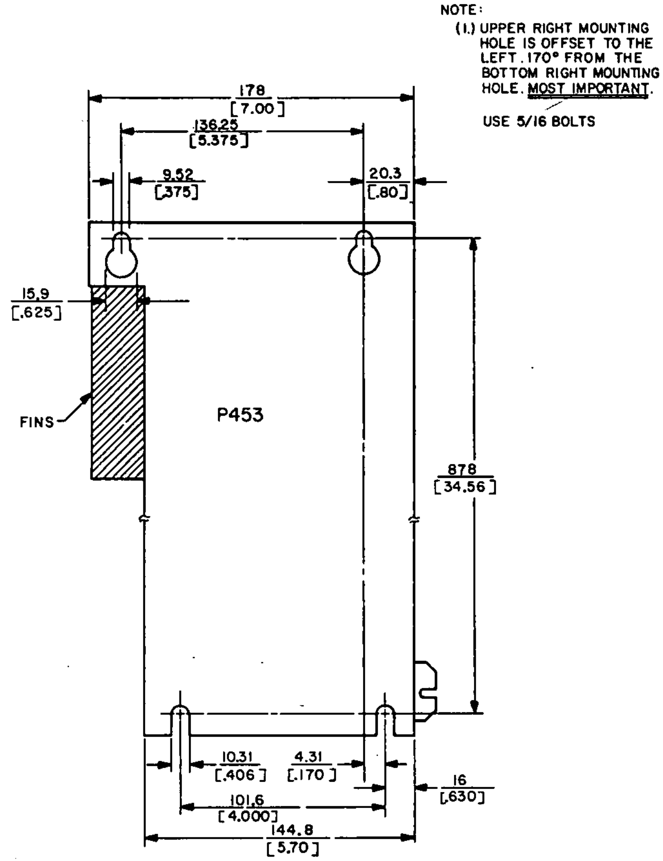


FIGURE 27(B) P453 MOUNTING DIMENSIONS

CAUTION: The user should insure that ground currents are not conducted along the system's cable run. The cable shield is grounded at the J200 Expander upon installation. The user should be especially wary of grounds which may occur at taps, connector supports, metal framework or other similar points.

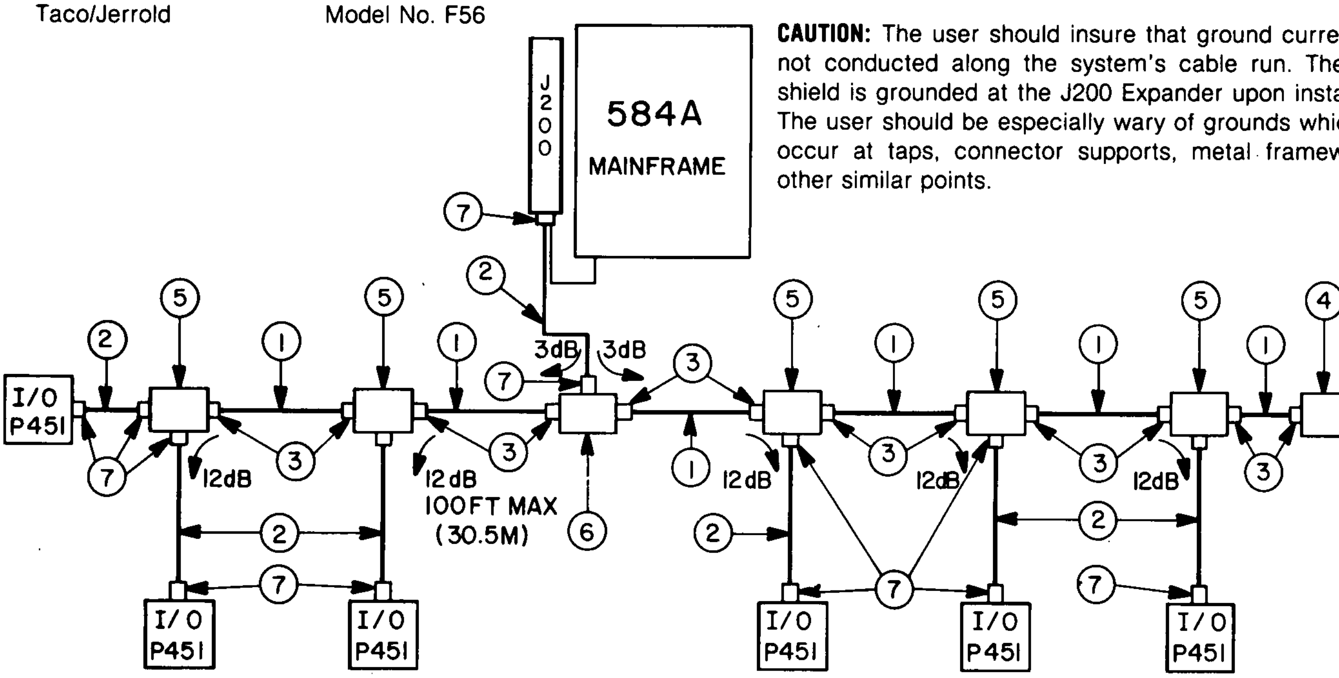


FIGURE 28 J200 REMOTE I/O CONFIGURATIONS

472

Section III — Basic Control Design

3.0 Introduction

This section discusses the following information:

1. Important machine concepts which, omitting detailed internal circuitry or design, will enable the user to better understand subsequent descriptions of the 584 Controller's functions.
2. A detailed discussion of each basic logic function (relay, timer, counter, and arithmetic functions).
3. A detailed discussion of optional capabilities.
4. A brief description of use of the P190 Programming Panel, which is required for entry or altering stored logic and data.

3.1 Important Machine Concepts

3.1.1 Controller Reference Numbers

Throughout the programmming of any 584 Controller, five-digit reference numbers are utilized to build the user's logic. These references are divided into two broad categories: discrete and registers. Discrete references are used for individual items that are either ON or OFF, such as limit switches, pushbuttons, relay contacts, motor starters, relay coils, solenoid values, etc. Register references are used to store numerical values such as counts, times, analog values, etc. All register references are values up to four BCD digits long (maximum value 9999).

Only four reference types are required to program the 584 Controller. Any specific reference can be used as many times as required by the particular application; there are no limitations on the number of times a reference is used. References are defined as follows:

0XXXX	-	coils/discrete outputs
1XXXX	-	discrete inputs
3XXXX	-	input registers
4XXXX	-	holding/output registers

NOTE: Except for the addition of a fifth digit, these reference numbers are the same as those used in MODICON's 184, 384, and 484 Controllers. To maintain this compatibility, references beginning with the digits 2 (2XXXX) are reserved for future use.

The address of each I/O housing (see Section 2.3) is very important in establishing proper reference for each I/O module. This depends upon the I/O system type and channel assignment. If 500 Series I/O is used with a J540 Adapter in a 200 Series Channel, use appropriate references assigned to the 500 Series I/O.

NOTE: The assignments shown in Table 9 are typical references. The I/O assignments can be altered to fit the application, usually by the addition of register I/O.

As can be seen from Table 9, discrete references begin at the top circuit of module one in housing one of the first channel, and continue in sequence through the housings and channels. Discrete references continue to the limit established by the system configuration. Table 9 lists the maximum of 32 channels of I/O, although a smaller limit may exist on a particular system. Typically, after discrete references are allocated, numerical I/O references are assigned. There is no requirement for the same number of discrete inputs as discrete outputs, although most systems will have an equal number of these references.

outputs, although most systems will have an equal number of these references.

TABLE 9A
CHANNEL ONE

200 Series Index Pin	500 Series Housing/Mod	References Input Module	Output Module
1	1-1	10001-10004	00001-00004
	1-2	10005-10008	00005-00008
	1-3	10009-10012	00009-00012
	1-4	10013-10016	00013-00016
2	1-5	10017-10020	00017-00020
	1-6	10021-10024	00021-00024
	1-7	10025-10028	00025-00028
	1-8	10029-00032	00029-00032
3	2-1	10033-10036	00033-00036
	2-2	10037-10040	00037-00040
	2-3	10041-10044	00041-00044
	2-4	10045-10048	00045-00048
4	2-5	10049-00052	00049-00052
	2-6	10053-10056	00053-00056
	2-7	10057-10060	00057-00060
	2-8	10061-10064	00061-00064
5	3-1	10065-10068	00065-00068
	3-2	10069-10072	00069-00072
	3-3	10073-10076	00073-00076
	3-4	10077-10080	00077-00080
6	3-5	10081-10084	00081-00084
	3-6	10085-10088	00085-00088
	3-7	10089-10092	00089-00092
	3-8	10093-10096	00093-00096
7	4-1	10097-10100	00097-00100
	4-2	10101-10104	00101-00104
	4-3	10105-10108	00105-00108
	4-4	10109-10112	00109-00112
8	4-5	10113-10116	00113-00116
	4-6	10117-10120	00117-00120
	4-7	10121-10124	00121-00124
	4-8	10125-10128	00125-00128

TABLE 9B
CHANNEL TWO

200 Series Index Pin	500 Series Housing/Mod	References Input Module	Output Module
1	1-1	10129-10132	00129-00132
	1-2	10133-10136	00133-00136
	1-3	10137-10140	00137-00140
	1-4	10141-10144	00141-00144
2	1-5	10145-10148	00145-00148
	1-6	10149-10152	00149-00152
	1-7	10153-10156	00153-00156
	1-8	10157-10160	00157-00160
3	2-1	10161-10164	00161-00164
	2-2	10165-10168	00165-00168
	2-3	10169-10172	00169-00172
	2-4	10173-10176	00173-00176
4	2-5	10177-10180	00177-00180
	2-6	10181-10184	00181-00184
	2-7	10185-10188	00185-00188
	2-8	10189-10192	00189-00192

473 over

CHANNEL TWO (Continued)

200 Series Index Pin	500 Series Housing/Mod	References Input Module	Output Module
5	3-1	10193-10196	00193-00196
	3-2	10197-10200	00197-00200
	3-3	10201-10204	00201-00204
	3-4	10205-10208	00205-00208
6	3-5	10209-10212	00209-00212
	3-6	10213-10216	00213-00216
	3-7	10217-10220	00217-00220
	3-8	10221-10224	00221-00224
7	4-1	10225-10228	00225-00228
	4-2	10229-10232	00229-00232
	4-3	10233-10236	00233-00236
	4-4	10237-10240	00237-00240
8	4-5	10241-10244	00241-00244
	4-6	10245-10248	00245-00248
	4-7	10249-10252	00249-00252
	4-8	10253-10256	00253-00256

TABLE 9C

CHANNEL THREE

200 Series Index Pin	500 Series Housing/Mod	References Input Module	Output Module
1	1-1	10257-10260	00257-00260
	1-2	10261-10264	00261-00264
	1-3	10265-10268	00265-00268
	1-4	10269-10272	00269-00272
2	1-5	10273-10276	00273-00276
	1-6	10277-10280	00277-00280
	1-7	10281-10284	00281-00284
	1-8	10285-10288	00285-00288
3	2-1	10289-10292	00289-00292
	2-2	10293-10296	00293-00296
	2-3	10297-10300	00297-00300
	2-4	10301-10304	00301-00304
4	2-5	10305-10308	00305-00308
	2-6	10309-10312	00309-00312
	2-7	10313-10316	00313-00316
	2-8	10317-10320	00317-00320
5	3-1	10321-10324	00321-00324
	3-2	10325-10328	00325-00328
	3-3	10329-10332	00329-00332
	3-4	10333-10336	00333-00336
6	3-5	10337-10340	00337-00340
	3-6	10341-10344	00341-00344
	3-7	10345-10348	00345-00348
	3-8	10349-10352	00349-00352
7	4-1	10353-10356	00353-00356
	4-2	10357-10360	00357-00360
	4-3	10361-10364	00361-00364
	4-4	10365-10368	00365-00368
8	4-5	10369-10372	00369-00372
	4-6	10373-10376	00373-00376
	4-7	10377-10380	00377-00380
	4-8	10381-10384	00381-00384

TABLE 9D

CHANNEL FOUR

200 Series Index Pin	500 Series Housing/Mod	References Input Module	Output Module
1	1-1	10385-10388	00385-00388
	1-2	10389-10392	00389-00392
	1-3	10393-10396	00393-00396
	1-4	10397-10400	00397-00400
2	1-5	10401-10404	00401-00404
	1-6	10405-10408	00405-00408
	1-7	10409-10412	00409-00412
	1-8	10413-10416	00413-00416
3	2-1	10417-10420	00417-00420
	2-2	10421-10424	00421-00424
	2-3	10425-10428	00425-00428
	2-4	10429-10432	00429-00432
4	2-5	10433-10436	00433-00436
	2-6	10437-10440	00437-00440
	2-7	10441-10444	00441-00444
	2-8	10445-10448	00445-00448
5	3-1	10449-10452	00449-00452
	3-2	10453-10456	00453-00456
	3-3	10457-10460	00457-00460
	3-4	10461-10464	00461-00464
6	3-5	10465-10468	00465-00468
	3-6	10469-10472	10469-10472
	3-7	10473-10476	00473-00476
	3-8	10477-10480	00477-00480
7	4-1	10481-10484	00481-00484
	4-2	10485-10488	00485-00488
	4-3	10489-10492	00489-00492
	4-4	10493-10496	00493-00496
8	4-5	10497-10500	00497-00500
	4-6	10501-10504	00501-00504
	4-7	10505-10508	00505-00508
	4-8	10509-10512	00509-00512

Since the I/O reference numbers are established by the index pin or housing/rack address, the I/O modules can be placed in any configuration provided they are properly addressed. Only the references required need be placed in the I/O configuration; inputs do not have to start at 10001 nor do outputs have to begin at coil 00001. Different types of I/O modules can be intermixed in any channel without regard to the voltage type. Output modules do NOT have to be installed to use coil references within the controller.

3.1.2 Multi-node Programming Format

As previously discussed, MODICON's 584 Controller controls the user's equipment by a program stored in its memory operating in conjunction with the I/O section. In block diagram this can be illustrated as shown in Figure 29.

474

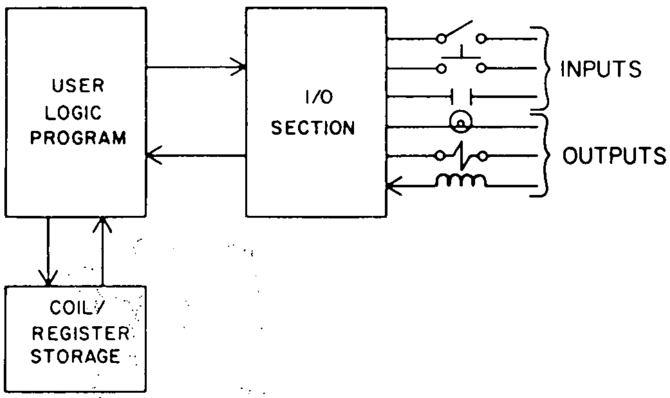


FIGURE 29 584 OPERATION BLOCK DIAGRAM

The multinode format allows for up to ten elements of the program in each horizontal rung of the ladder diagram. Up to seven of these rungs can be combined into a network of relay contacts and other programmming elements (timers, counters, etc.); each network can have up to seven coils placed at the extreme right of the network. Coils are displayed in the eleventh column which is reserved solely for logic coils.

The network serves as one basic parameter of the ladder diagram program. It is defined as a group of program elements comprising from one to seven rungs, each with up to ten elements which are interconnected. The quantity of networks or logic elements that can be entered depend upon the memory size utilized and the complexity of the network. Each programmed element can require as little as one word of memory. This results in extremely efficient memory utilization for smaller applications. See Section 5.1.3 for exact details on memory utilization.

NOTE: The P190 Programming Panel will display in real time the amount of memory available for additional logic, as well as memory used by existing logic.

The basic programming element is the relay contact, as shown in Figure 31. The contact can be either normally open or normally closed; the branch to the next lower rung is optional. The five digit reference number below the contact controls the power flow which activates the contact (makes normally open contacts pass power and normally closed contacts block power flow). Within a network power flow will be allowed only vertically from left to right; it is never allowed to flow from right to left. When properly displayed on the P190 screen, power flow will be indicated for all relay contacts by intensifying those contacts that are passing power from left towards the right.

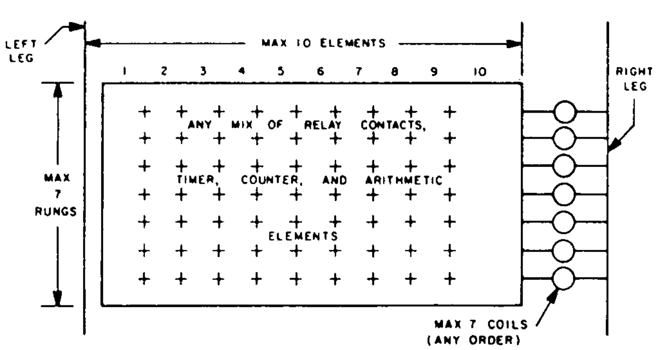


FIGURE 30 MULTI-NODE PROGRAM FORMAT

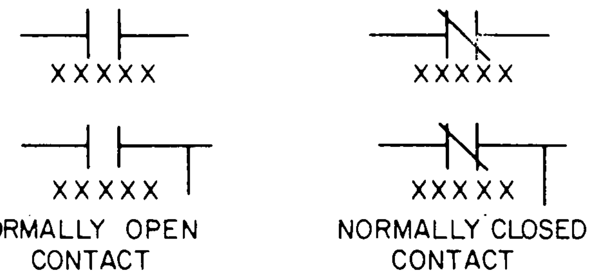


FIGURE 31 RELAY CONTACT TYPES

Whenever data is entered into the controller with the P190 Programmer (references, contact type, disable status, etc.), the data is entered directly into the controller's memory. If power should be interrupted prior to completion of the programming, whatever data has been entered will be retained. No additional processing is required, such as further assembly of the data. Networks can be changed either in whole or in part, or added and deleted at any time using the Programming Panel. In addition, any coil output or input may be tested by simulating outputs and inputs with the disable feature.

3.1.3 Scan

The 584 Controller examines (solves) each network of interconnected logic elements in the networks' numerical sequence. Network one is the first network to be solved on each scan, followed by network two, three, etc., until all available networks are solved. The controller then turns to network one. This fixed scanning occurs at an extremely rapid rate from the time power is applied to the processor until power is removed. Within each network, logic elements are solved by columns from the left rail to the right rail (where coils are displayed) and from top to bottom within each column.

Each network's result is immediately available to all subsequent networks regardless of whether this result is a change in coil state or a change in numerical value. Networks are solved in order of their numerical number and NOT by the numerical value assigned to any coil.

During scan of the programmed logic, input modules are read and outputs driven. At specific points during the scan, inputs will be updated and made available to the user's program, as well as new coil status provided to drive outputs. Inputs are updated and coils provided (I/O exchanged) at the rate of two channels (256 input points and 256 output points) per exchange. Between each exchange, a portion of the total logic is solved; this portion is defined as "one segment". The quantity of logic in a segment depends upon user requirements since the user can select the segment in which each network is programmed. The total quantity of segments is equal to the quantity of I/O channels divided by two. Figure 32 illustrates schematically the total scan pattern for a controller with four channels of I/O.

3.1.4 Memory Protect

The 584 Controller is provided with a Memory Protect Key designed to prevent accidental or unathorized changes to the

475 over

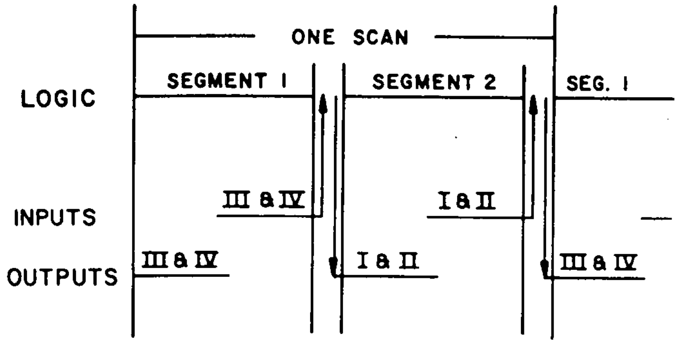


FIGURE 32 TYPICAL SCAN PATTERN

memory. When the memory protect keylock switch (see Figure 8) is placed in the ON position, the user's logic cannot be altered by any external device, such as the P190 Programmer, Tape Loader, Telephone Interface, or a Computer. The logic can only be examined, not altered. Thus, by placing memory protect ON and removing the key, maintenance personnel can use the programmer to monitor the system, but they cannot make unauthorized changes. Only specific personnel who are provided access to the key can change the system.

NOTE: The Memory Protect feature protects the user's logic, but does not protect those elements that normally change -such as registers and I/O status.

3.1.5 Disable/Enable

To simplify the checkout and maintenance of a control system, a disable/enable function is incorporated into all 584 Controllers. Any logic coil selected by the CRT's cursor can be disconnected from its logic by depressing the DISABLE pushbutton. If the coil was OFF when the pushbutton was depressed, it will remain OFF; if it was ON, it will remain ON. The coil is no longer controlled by the program in the controller, but by the operator of the programming panel. The coil can be cycled ON/OFF/ON/OFF by successively depressing the FORCE pushbutton. The Disable status may be changed only if Memory Protect is OFF.

When disabled, the logic coil, all references to this coil in the ladder diagram, and any outputs driven from this coil, will be affected solely by the disable condition. The internal programmed logic still remains in the controller and will re-establish control when the coil is enabled; however, this internal logic has been completely bypassed for this coil by the disable function. The disable status of any coil is fixed until altered by a programming device. New networks can be displayed, other coils disabled, power interrupted, memory protect turned ON, or any other change made to the system without affecting the disable status of any coils. Any coil disabled either OFF or ON will retain that state until changed by a programming device.

NOTE: To re-enable a logic coil, the ENABLE pushbutton is depressed while the cursor is on that coil.

In addition to logic coils, discrete inputs can also be disabled. The cursor is moved out of the logic area (below the seventh rung) and the five digit reference number for the desired input is entered. The GET key is pressed followed by the DISABLE pushbutton. This action removes control of that input from the "real world" and assigns that control to the operator via the programming panel. The input can be forced either ON or OFF, and all logic that uses this discrete input will respond to the disable status not the real world. The disable status is permanent and can be altered only by programming devices with memory protect OFF. At any one time, as many logic coils and discrete inputs as desired can be disabled either ON or OFF.

NOTE: DX functions ignore DISABLES.

NOTE: A coil once disabled will remain in that state until reenabled from either the P190 or the 584 RAP Panel. A record should be maintained of all disabled logic coils and inputs so that they can be enabled at a later date. A search operation (standard function with P190) or ladder listing will show the disable state of any logic coil or input.

In checking a system, the disable function can be used to verify the proper wiring and operation of all discrete outputs. Each output is displayed in a network on the P190 Programming Panel and then disabled. The coil can be cycled ON-OFF-ON-OFF, etc., with proper operation of the discrete device being observed. It is recommended that the logic coil be enabled before the next output is tested to prevent an undesirable disable status from occurring.

CAUTION: Insure that disabling outputs or inputs do not create hazardous machine or process operation.

If an input such as a limit switch fails to operate properly, its effect can be temporarily simulated by disabling the input and forcing it to the desired state (ON or OFF). This is particularly useful if the input is preventing the control system from functioning. Another use for disabling inputs is to simulate the operation of the control system prior to connecting the I/O. However, since the disable feature for either input or logic coils is a powerful function which causes catastrophic results if improperly used, the keylock memory protect can be used to ensure that changes of the disable states be made only by authorized personnel.

3.2 Basic Programming

All 584 Controllers are provided with the capability to program or simulate the functions of relays, timers, counters, and other functions discussed in Sections 3.3 and 3.4. All programming is done on the basic format of up to ten elements in each horizontal row or rung, and up to seven of these rungs connected to form a network. Any network can be a single rung, two rungs, or up to seven rungs as long as there is some connection between the elements of each rung. This connection can be as simple as the left power rail of the ladder diagram. Each network can have up to seven coils, located to the extreme right of any/all rungs of the network. These coils can be assigned any valid logic coil number available in the controller.

NOTE: Logic coil reference numbers can be used as coils only once. This includes the "B" element of DX Move and Matrix Functions.

The quantity of logic coils, discrete inputs, programmed logic elements, numerical storage locations (registers), etc. depends upon function level and memory size. Function level one allows up to 2048 total discrete references (coils and inputs summed in any mix each divisable by 16) and up to 1920 total register references (input and holding as discussed in Section 3.3). Level one and three use one word per programmed element and are suitable for normal size applications. Function level four allows a maximum total of 8192 discrete references and a

476

total of 9999 register references. This capability is referred to as "extended references". To address such a large quantity of logic signals and data storage, one and one-half words is required per data storage and programmed element. Level four is suitable for very large relay logic applications or large data handling/report generation/maintenance diagnostic applications. Many individual configurations are available for each memory size. Several examples are provided in Table 10; for additional details, see Section VI.

TABLE 10
SAMPLE MEMORY CONFIGURATIONS

Size (16 Bit Words)	Avail. Ref.	Logic Coils	Discrete Inputs	Input Reg.	Holding Reg.	Logic Space	Avg. Elements
4K	Basic	352	512	32	86	3,500	2,800
8K	Basic	704	1024	16	266	7,200	5,800
8K	Basic	512	512	32	1,724	5,700	4,600
12K	Basic	1024	1024	64	391	11,000	8,800
12K	Basic	1024	1024	64	1,856	9,500	7,600
12K	Ext.	1024	1024	64	5,290	5,500	2,900
16K	Basic	1024	1024	64	448	15,000	12,000
16K	Ext.	1536	2048	16	651	14,500	7,700
16K	Ext.	1536	1024	64	6,570	8,000	4,250
24K	Ext.	2560	2048	16	715	22,300	11,800
24K	Ext.	1536	1024	64	8,821	13,600	7,200
32K	Ext.	3072	2048	16	984	30,000	16,000
32K	Ext.	1536	1024	64	9,000	20,382	10,800

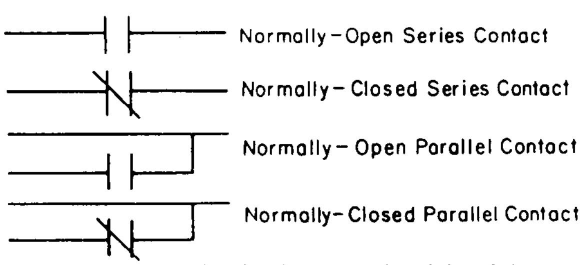
NOTE: Average elements represent the total quantity of relay contacts, coils, and non-relay references that can be programmed from logic space available from a typical network. This value attempts to compensate for a variety of memory requirements.

3.2.1 Relays

When programming a relay contact into the general format (see Figure 37), any horizontal arrangement of contacts can be used.

NOTE: Contacts can NOT be placed vertically.

Most relay programming will utilize Normally Open (NO) or Normally Closed (NC) contacts with the following symbology:



The vertical connection is always to the right of the contact with which it is programmed and connects to the next lower rung. Vertical connections can be added or deleted at any time and do not require memory space. They can be added to any logic element, relay or non-relay function; however, they can not be programmed with logic elements in the last (7th) rung of a network or from coils. Power can only flow from the left power rail towards the right, or through vertical connection (up or down); power flow is NOT possible in the reverse direction (right to left).

Logic coils (0XXXX references) can be internal (used only within the controller) or output (used internally as well as control an output circuit). All coils used by the controller exist within the controller; output modules need not be installed to program a coil. Normally, the coils that control outputs are assigned the lower reference numbers and internal coils assigned the higher references. For example, if 256 outputs are desired of a total of 352 coils, the output coils are references 00001 - 00256. References 00257 - 00352 are internal coils. However, the 584 Controller system has the capability to output any valid coil, in groups of 16 with the last (highest) reference evenly divisable by 16, up to the limit of I/O section capacity.

A logic coil (output or internal) can be used as a coil only once; however, references to contacts controlled by that coil can be used as many times as required. There is no limit to how many times any reference (0XXXX-coils, 1XXXX-discrete inputs, 3XXXX-input registers, and 4XXXX-holding registers) is used in a program. Output coils that are not used to drive discrete outputs (e.g., no output module assigned to that address in the I/O section), can still be used as coils in programming. Thus, any unused output coil can be used for internal functions the same as internal coils are used.

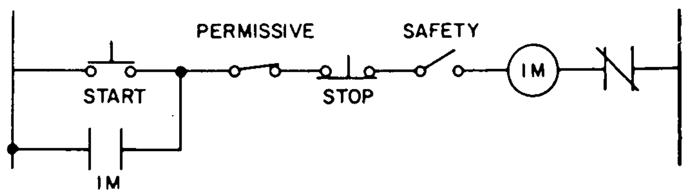


FIGURE 33 SAMPLE RELAY LOGIC

If the logic in Figure 33 were to be implemented in the 584 Controller, the control elements must be connected to input circuits in the I/O configuration and appropriate outputs assigned. Any available input modules of the proper voltage level can be used; Figure 34 illustrates assumed input assignments and wiring details. Output number 12 is assigned to operate the external device. The resultant internal logic which is programmed by the user is shown in Figure 35.

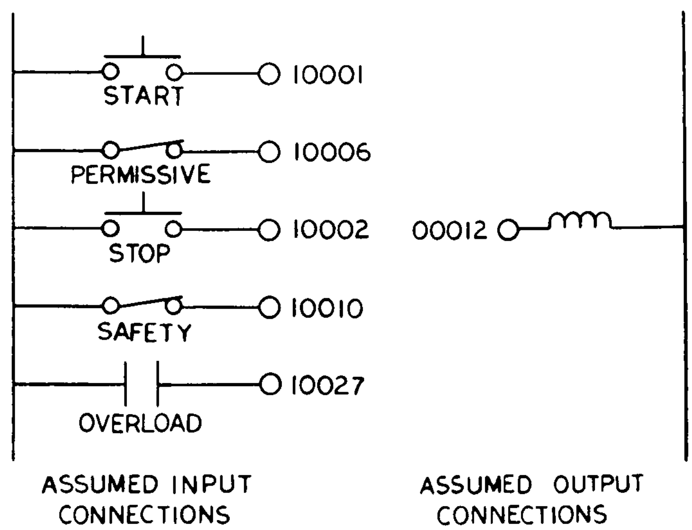


FIGURE 34 ASSUMED I/O CONNECTIONS

477 over

NOTE: Any input that is wired normally closed (e.g. 10002 and 10027) will be normally energized (ON reference). When used in the logic, the input should be programmed as a normally-open contact which will allow power to flow unless the input is de-energized.

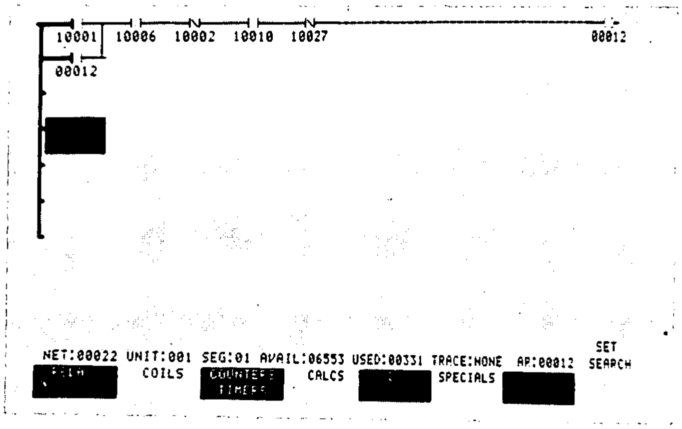


FIGURE 35 SAMPLE PROGRAMMED LOGIC

Retentive Coils:

Any logic coil can be programmed to retain its previous state (ON or OFF) upon system start-up. All coils that are not selected to be retentive will be de-energized when power is restored (power-up). Retentive coils can be used to retain an ON or OFF condition of a logical function similar to latching relays. If a retentive coil was ON prior to the power failure, it will be returned to the ON state when power is returned regardless of how long power was lost. Of course, if the coil was OFF, it will remain OFF when power is restored.

To create a retentive coil, its coil symbol is changed to -(L)- in lieu of the normal -( )-. No relay contacts need be programmed to implement retentive functions, nor is additional memory required. As soon as a logic coil is energized or de-energized, its retentive state is established. All logic coils (output or internal) can be made retentive merely by selecting the appropriate coil symbol. If the logic in Figure 35 was to be retentive, it would be programmed as shown in Figure 36.

NOTE: Once power is restored, retentive coils will follow the state (ON or OFF) as dictated by the logic controlling that coil.

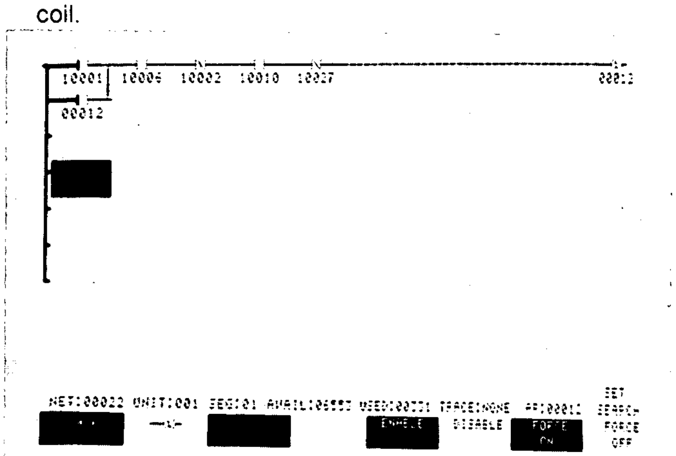


FIGURE 36 SAMPLE RETENTIVE LOGIC

Extended Logic

If more than 10 elements are required in a rung to satisfy a complex control function, an internal coil can be used to represent a partial result. A contact referenced to this coil is then placed as the first element in another rung and additional contacts are entered into this second rung. The coil of this second rung can be an output that represents the resultant logic of up to 19 series elements or an internal coil for further extension of the logic. There is no limit to how many times the logic can be sequenced or "cascaded" in this manner. However, keep in mind that the quantity of logic coils and elements is established by memory size. Internal coils can also be used with up to ten elements to represent a single block of logic which is used repetitively in the program. As an example of extended logic, refer to Figure 37.

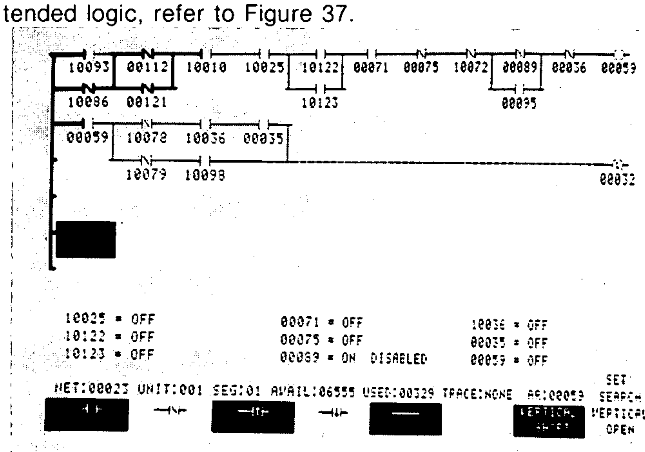


FIGURE 37 EXTENDED LOGIC PROGRAM

Oscillator

In many situations a signal is required that oscillates ON-OFF-ON-OFF, changing states each scan thus completing a cycle every two scans. This signal can be used to flash lights or cause some action to be taken every other scan. To create such a signal using logic coil references, a single normally closed contact is programmed to control the coil. The contact is then referenced to the coil. Figure 38 illustrates such an oscillator using coil 00464.

NOTE: 184/384 users will recognize this as "Watchdog" timer logic.

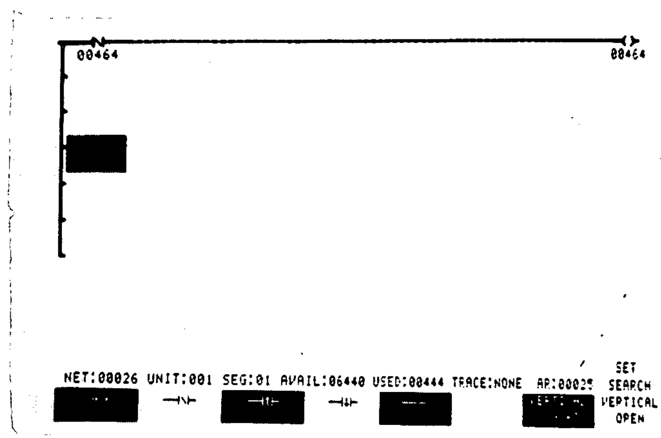


FIGURE 38 SAMPLE OSCILLATOR

478

Fault Indications

If an external indication is required when the processor shuts down, an output can be disabled ON. This output will be ON as long as the processor is running. It will be OFF if power is removed or if a processor error has been detected and automatic shut down accomplished. However, since someone may consider a diasbled output as a mistake, a simple circuit referencing either the oscillator coil or the output coil can be used to "document" the output as a valid operation. Two contacts in parallel, one normally open and the other normally closed, are used as shown in Figure 39; only one of the two methods shown in this figure need be implemented.

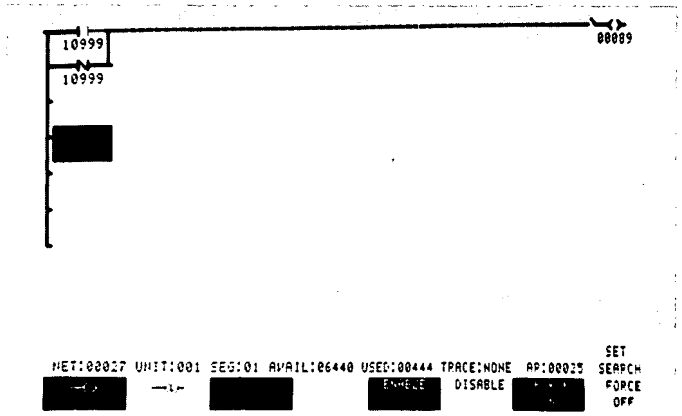


FIGURE 39 SAMPLE FAULT INDICATOR

NOTE: The output will also be OFF if the output module should fail.

Critical output modules can be monitored by maintaining an ON output on that module to an input circuit; the input will always be ON unless the output or input modules should fail. Critical input modules can be monitored by wiring an input on that module directly to power. The input will still always be ON unless the input module should fail.

First Scan Indicator

In some applications, action must be taken on the first scan following a power failure to clear registers, outputs, etc. After the first scan, the elements that were cleared are controlled by other logic - the clear control is eliminated. The logic displayed in Figure 40 should be programmed near the entire logic's conclusion or, at least, after all clear operations have been performed. For the first scan only, all normal (non-retentive) coil references will be OFF until their respective logic is solved; in this example, normally closed references to coil 00420 will pass power only on the first scan.

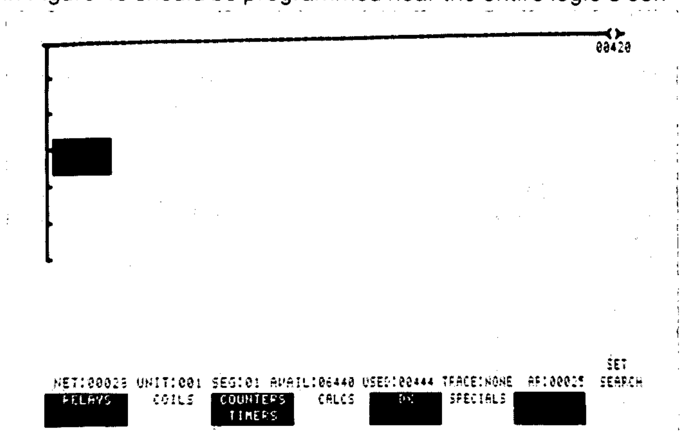
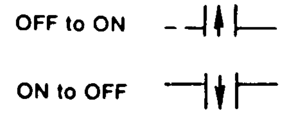


FIGURE 40 SAMPLE FIRST SCAN INDICATOR



Transitional Contacts

In addition to conventional normally open and normally closed contacts, all 584 Controllers can utilize transitional contacts.

These transitional contacts can be used anywhere in networks where the conventional NO and NC contacts previously occurred. They will pass power for exactly one scan whenever the discrete signal or signal to which they are referenced transitions from either OFF to ON or ON to OFF (depending upon transitional type selected).

Transitional contacts can be referenced to any input or logic coil (output or internal). For an example refer to Figure 42. When input 10062 is energized and all permissives satisfied, coil 00007 will be energized for one scan. If the permissives are not satisfied when input 10062 is energized, coil 00007 will not be energized even if the permissives are later satisifed.

NOTE: 184/384 users will recognize these as "one-shots".

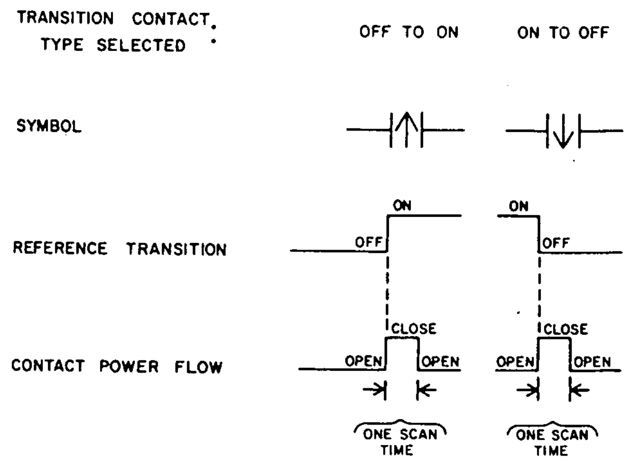


FIGURE 41 TRANSITIONAL CONTACTS

Programming with the P190 Programmer

Use the following steps to enter logic with the P190 Programmer (assuming that the programmer has been connected to the 584 service connector). For details on P190 Programmer installation, connection, and operation, see Appendix A.

1. Depress START NEXT. Verify that the left leg has been created for construction of the new network. The cursor will be at the location where the first element is to be entered.
2. Enter the reference number into the Assembly Register (AR) at the lower right of the CRT screen. Depress the Normally Open (-| |-) dynamic pushbutton, entering both the contact type and its reference from the AR. Depress the Vertical Shunt-dynamic pushbutton to add the vertical short to the right of the contact.

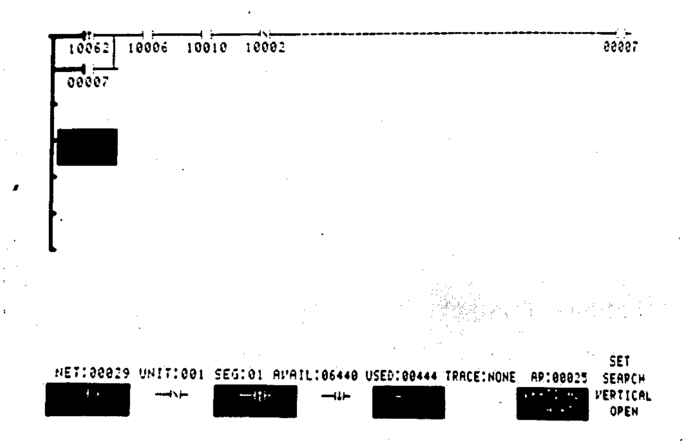


FIGURE 42 SAMPLE TRANSITIONAL LOGIC

NOTE: Errors are easily corrected by reentering the desired values.

3. To correct an element already in the network, place the cursor on the element to be changed. Reenter the correct element.
4. Move the cursor to the next element location in the network. Elements can be entered in any order desired; networks do not have to start with the upper left element.
5. Repeat steps 2, 3, and 4 for other elements in the network. After all elements are entered, the coil is entered and labeled with any unused output or internal logic coil.

NOTE: The CRT will automatically display horizontal shunts between the last programmed element and the coil in the extreme right column. These shunts will appear as dotted lines and do not require memory locations in the 584 Controller. If these shunts were programmed by the user, they would require memory.

3.2.2 Timers

Timers can be placed anywhere in a network where sufficient space exists. They are built vertically and require two non-relay elements, one on top, the other on the bottom. The general form of a timer is shown in Figure 43. Within all 584 Controllers are three crystal-controlled clock signals which drive all timers. Any timer can be programmed to respond to the seconds clock, 1/10 second clock, or 1/100 second clock; the only limitations to how many timers are referenced to any one clock signal are the user memory size and the number of available holding registers.

Referring to the typical timer in Figure 43, there are two inputs for entry of power flow located to the left of the timer. The upper input controls when the timer increments time, and the lower input controls when the timer is reset to zero or enabled. The timer is enabled when the lower node receives power flow, reset when no power flow is available. The upper element of the timer contains the preset value which limits the timer's maximum value. This preset can be a fixed value of three digits (001 to 999), which represents up to 999 seconds (timer in seconds), 99.9 seconds (timer in tenths of seconds), or 9.99 seconds (timer in hundredths of second).

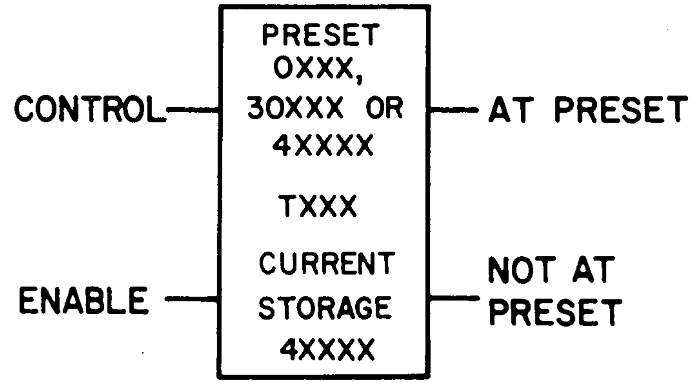


FIGURE 43 TIMER/COUNTER GENERAL FORM

NOTE: Controllers with extended references can utilize fixed presets of up to four digits (maximum 9999).

The lower element refers to a register within the controller where the current time is maintained. In the center of the timer is a display to indicate the rate at which that particular timer is programmed to operate (T1.0 = seconds, T0.1 = tenths of seconds, and T.01 hundredths of second).

On the right of the timer are two nodes from which the logical output (i.e., coil) of the timer is available. These nodes will provide power to any contacts, shunts, coil etc. programmed to the right of the timer. The upper node provides power only when the timer is at its preset value. This output de-energizes and stops providing power whenever the lower input on the left (reset) stops receiving power. Whenever this output is energized, the timer stops and no further incrementing of time beyond the preset is possible. The lower node provides power whenever the timer is NOT at its preset. This output will stop passing power when the timer reaches its preset.

NOTE: During the normal operation of a timer, this function will stop accumulating time when the accumulator register equals the preset. Thus a timer will not allow the accumulator to exceed the preset. If the accumulator is forced to exceed the preset (by programming a value in the accumulator which is larger than the preset), the 584 Controller will set the accumulator equal to the preset.

When the timer is enabled (lower input on left receiving power), the timer will accumulate time when the upper left input receives power. The upper signal can be turned ON-OFF-ON as many times as necessary and the timer will accumulate the amount of time the signal was ON up to the preset value. Each time the upper input is re-energized, time begins to accumulate from the previous value stored in the accumulator (regardless of how long the signal was OFF). Timers reset only by the lower input signal, and they are completely retentive on power failure. Whenever the reset signal (lower left input) is de-energized regardless of the time value, the timer will be reset to zero and held at that value until the reset signal is energized.

Specifying Time

The upper element (preset) of the timer is reserved for entering the desired amount of time, in seconds, tenths of seconds, or hundredths of seconds to which the timer is to time. If the preset has a value of 009 it will represent 9 seconds (T1.0 Timer), 0.9 seconds (T0.1 Timer) or 0.09 seconds (T.01 Timer). This reference can be to a fixed value up to 999 or a

480

register. When referenced to a register value (30XXX or 40XXX), the content of that register is used as the preset value up to four digits (maximum value 9999).

NOTE: The timer's upper element is reserved to hold the timer's designated limit.

Short-term timer accuracy is timer rate or one scan time (whichever is larger).

Storing Time

The number entered into the lower element of the timer must be a storage register (reference 4XXXX) where the current time is stored. Registers are locations within the memory where numerical values up to four digits (9999) are stored. A simple definition of registers are a "mailbox" or "bucket" where information (in this case, time) is permanently stored. The registers are referred to or named by reference numbers beginning with 40001 and continue consecutively to the maximum established by the controller's configuration.

NOTE: Registers are inherently retentive upon power failure.

Normally, each timer will have its own holding register selected by the user. Thus, the maximum number of timers available with each configuration is the quantity of holding registers. Do not use a holding register as the storage location of more than one timer. In general, however, all such locations will not be used to store timing information, since these locations are useful for a great many other purposes, as will be seen later. The contents of any storage location may easily be examined.

The timers in the 584 Controller are ON-delay energizing timers. To obtain other types such as OFF-delay energizing, ON-delay de-energizing, or OFF-delay de-energizing, the programming is adjusted to obtain the desired result. For OFF-delaying action, the energizing logic (upper input, left side) is programmed with reverse logic, i.e., normally closed contacts instead of normally open contacts. To have de-energizing action, the lower output of the timer is used in lieu of the upper output.

In addition, a vertical short, or shunt, can be connected between the two inputs of any timer. With such a shunt, the timer is both energized and enabled by the same logic; thus, a timer will not accumulate time, and it is reset to zero whenever it stops timing. Figures 46-49 illustrate four types of timers, none of which are accumulative. Input 10053 is used to start the timers, and coil 00019 is the output reflecting the desired timing signal. Of course, additional logic can be added to either input 10053 or the output or both (up to ten elements per horizontal rung). Figure 50 illustrates the timing relationship of the various types of timers shown in Figures 46-49.

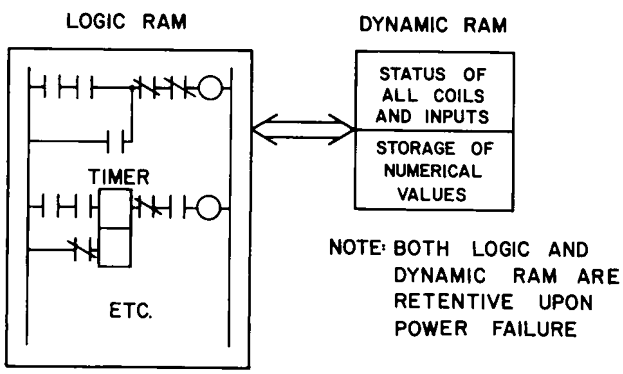


FIGURE 44 RELATIONSHIP BETWEEN LOGIC AND DYNAMIC RAM

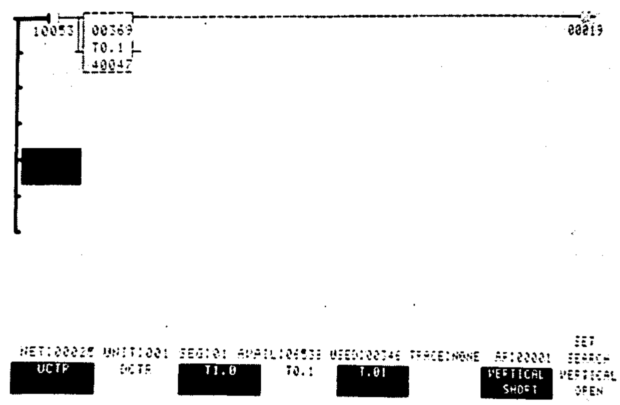


FIGURE 46 ON DELAY ENERGIZING TIMER

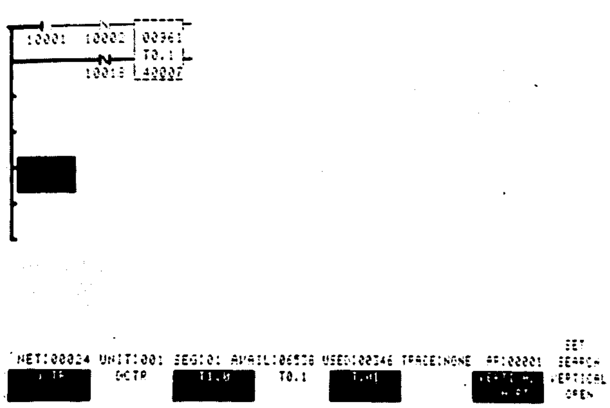


FIGURE 45 SAMPLE TIMER LOGIC

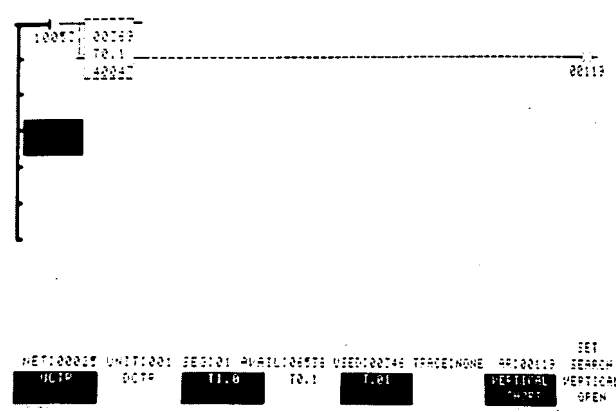


FIGURE 47 ON DELAY DE-ENERGIZING TIMER

481 over

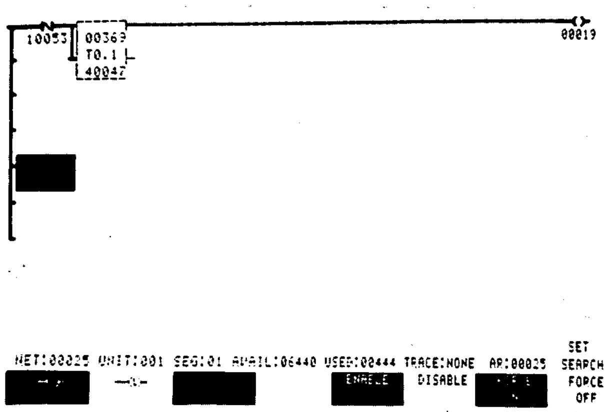


FIGURE 48 OFF DELAY ENERGIZING TIMER

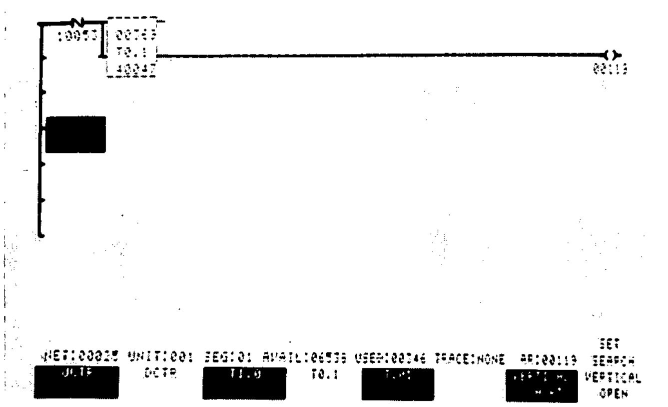


FIGURE 49 OFF DELAY DE-ENERGIZING TIMER

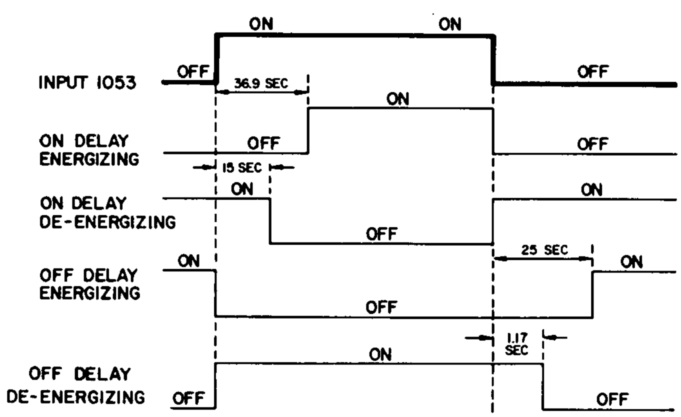


FIGURE 50 TIMING RELATIONSHIPS

3.2.3 Counters

Every 584 Controller has the capability to count signals in both accending (up counting) and decending (down counting) mode. These two separate counting functions operate similarly to the timing function except for the control input (upper left input). Either type counter will increment or decrement one count when the control input transitions from OFF to ON. Only one count will be recorded regardless of how long the signal was OFF or ON, or if the controller's operation was interrupted. The operation of each counter type is discussed in the following paragraphs.

NOTE: Individual holding registers should not be used by more than one timer or counter.

UP COUNTER

As with the timer, two vertical non-relay elements are used to program a counter, requiring only one of the ten columns in a network format. The inputs to the left (control and reset/enable) can be a single relay contact, multiple contacts in a network format, or the result of relay and non-relay elements. The upper input signal controls when the counter is incremented; the reset/enable input when not passing power resets the counter to zero.

The upper element represents the preset or limit on the counter and controls the status of the outputs. The preset can be a fixed value up to 999 or a register reference (30XXX or 4XXXX) whose content (maximum value 9999) is used as the preset. The counter can not exceed the preset value.

NOTE: Controllers with extended references can utilize fixed presets of up to four digits (maximum 9999).

The lower element must be a unique holding register (4XXXX reference) in which the current count is stored. The content of this register is incremented on each positive (OFF to ON) transition of the control (count) signal, but can not exceed the present value.

NOTE: Counters are retentive since holding registers are retentive upon power failure.

The outputs again are similar to the timer. The upper output supplies power if the count is at the preset; the lower output supplies power if the count is NOT at preset. Only one output will supply power flow. These outputs can be connected directly to coils, to relay contacts (with or without coils), to non-relay functions, or left unconnected.

DOWN COUNTER

The down counter is identical to the up counter, except that the count is decremented (decreased by one) when the control input transitions from OFF to ON. The reset/enable input, when NOT receiving power, resets the counter to the preset value; when the reset receives power flow, the counter is enabled and can respond to the count signal. The preset can be a fixed value or a register reference. The lower element must be a unique holding register; its content is set equal to the preset upon reset. The upper output supplies power flow at the count of zero and the lower output at non-zero counts. Instead of starting at zero and counting up (0,1,2,3, etc.), the down counter starts at the preset (e.g. 457) and counts down (457, 456, 455, 454, etc.).

NOTE: The holding register referenced in the lower element can be used in another up counter to provide up/down counting.

CASCADED COUNTER/TIMERS

Timers and counters can be interconnected or "cascaded" to satisfy any required logic. As many counters/timers as necessary can be placed in a network and are limited only by the 10 x 7 format. Two timers can be placed in series by using the upper output of one timer as the enable signal to a second timer. These two timers allow twice as much time to be measured; they in effect are additive. Timers and counters can also be cascaded to "multiply" their range. Figure 51 illustrates

482

a timer/counter network that produces a calendar measuring time in seconds, minutes, and hours. A calendar can be built in less than 100 words (including numerical storage) which automatically compensates for leap years and variable days per month.

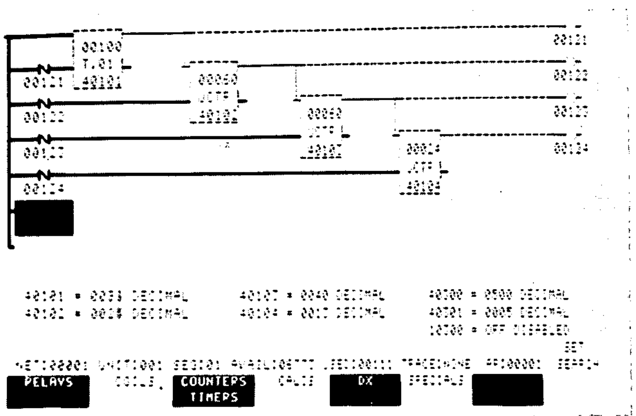


FIGURE 51 SAMPLE 24 HOUR CLOCK

ENTERING A TIMER OR COUNTER

To enter the logic in Figure 45 into a 584 Controller with the PI90 Programmer, take the following steps. It is assumed that memory protect is OFF and the Programmer has been properly connected (see Appendix A):

1. Depress START NEXT. Verify that the left leg is displayed indicating where the new network will be constructed. The cursor will be at the location where the first element is to be entered.
2. Enter the reference number 1001 into the Assembly Register (AR) at the lower right of the CRT screen. Depress relay dynamic pushbutton. Depress the Normally Open (⊣ ⊢) dynamic pushbutton, entering both the contact type and its reference from the AR.
3. The cursor is moved to the right. Enter the next element (10002) by repeating Step 2
4. Again move the cursor down and to the left into the first column second row. Depress horizontal short. Move the cursor one position to the right. Enter the reference 10018. Depress normally closed dynamic pushbutton. Move the cursor to the right one position and up one position. Enter the preset into AR. Depress the Change Node pushbutton.
5. Depress the Timer/Counter dynamic pushbutton and the type of timer desired. In this example, the T0.1 dynamic pushbutton is depressed to create a 1/10 second timer. The preset is entered from the AR. This procedure works for scan times under 100 msec.

   NOTE: The complete two element format for the timer is entered when the timer type is selected.
6. Move the cursor down, enter the value 40007 into the AR (4,0,0,0,7), and depress the ENTER pushbutton. The timer will use register 40007 to store its current time.

EXAMPLE I
SCAN TIME EVALUATOR

To review timer and counter operation, a network can be built to evaluate the average scan time of a controller while it is functioning. A sample of such a network is provided in Figure 52. References used in this example are general and can be altered to fit the application. Coil 300 is designed as an oscillator and cycles ON-OFF once every two scans. The counter, as soon as input 10300 is de-energized, will begin to count coil 300 up to 500 positive transitions. Five hundred counts are equal to 1000 scans.

When input 10300 is de-energized, the counter also enables the timer via its lower output. The timer measures how long it requires the counter to count 500 pulses (1000 scans) and will stop timing when the counter's preset is reached. The time (in seconds) for 1000 scans is in register 40301 when the counter reaches its preset. This time is also the average scan time in milliseconds. Slightly more accuracy is obtained by using a timer in tenths of seconds, which results in average scan time in tenths of milliseconds (e.g. 17.8 mSec).

NOTE: Memory is saved by not assigning coils to timer or counter outputs since they are optional and of no use in this application.

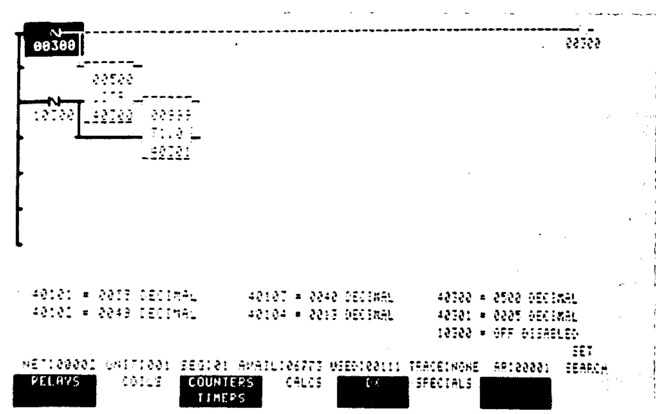


FIGURE 52 SCAN TIME EVALUATOR

3.3 Registers

The 584 Controller contains two types of storage for numerical values. They both have the same capacity, yet are used for two different purposes. The first are input registers that are reference numbers in the form 30XXX; the second are holding registers (reference 4XXXX). Each register requires one word of memory (1.5 words with extended references), and can store values up to 9999 (BCD format) or 16 bits (binary format). They can be displayed and altered from either a programming panel or the register access panel (RAP).

The exact quantity of each reference available with any particular controller is variable and depends upon memory size and the system configuration table. The system configuration table can be altered by either the 24 hour MODICON Service Center (using the Telephone Interface), or from the P190 Programmer with the 584 Utility Tape. Generally, there will be fewer input registers (typically 30-100) than holding registers (typically 200-9,000).

483 oves

Input registers are used to store numerical values received via the I/O section from external numerical devices such as thumbwheels, high speed counters, shaft encoders and analog signals (4-20 ma, 10-50 ma, 0-10 Vdc, 1-5 Vdc, -10 to 0 Vdc, etc.). These values are provided for reference purposes only (internal to the controller). They cannot be altered from inside the controller; their values are changed only by external devices. Typical wiring for an input register is provided in Figure 53 using a 200 series 16 circuit DC input module for illustrative purposes. 500 Series I/O modules can be used as long as they utilize the same relative addresses as the 200 series modules (see Table 9). After a power failure, input registers will be updated with new values from the I/O section prior to scanning any logic.

Holding registers are extremely valuable when used in the 584 Controller with non-relay logic. These registers are all retentive upon power failure and will change only when directed to do so by the logic or the operator. They are NOT assigned to only one network as are coils, and they can be shared by many logic functions. Thus, a counter's current count can be used as a preset on a timer or for another non-relay function. Within a register, the individual bits can be addressed by advanced (matrix) functions. Bit numbers are assigned from 1 at high-order end to 16 at low-order end (see Figure 72).

Output registers are special holding registers since they serve both to store a numerical value and to deliver that value as an output. Output registers, like holding registers, do NOT require any I/O hardware for storage location. I/O modules only add the capaiblity to receive or output numerical values. Although the 584 Controller can assign any holding register to be an output register, normally the lower reference numbers (e.g. 40001-40256) are used as output registers. Typical devices driven from output registers include LED numerical displays, and analog signals (4-20 ma, 1-5 Vdc, 0-10 Vdc, etc.).

Within the 584 Controller, all numerical data is stored as a 16 bit binary value and will be converted automatically to a decimal value when displayed or output. Similarly, input registers are converted to binary prior to storage in the controller.

All registers in the I/O section (both input and output) can be coded as BCD or Binary. If binary data is desired, each individual register can be separately coded for this data. Binary data is required when interfacing with 200 series analog I/O, as well as some encoder systems. Changes from BCD to Binary and vice versa can be accomplished by altering information stored in the system data table. Again, changes to this table can be made from either the Service Center or the P190 Programmer using the 584PC Utility Tape.

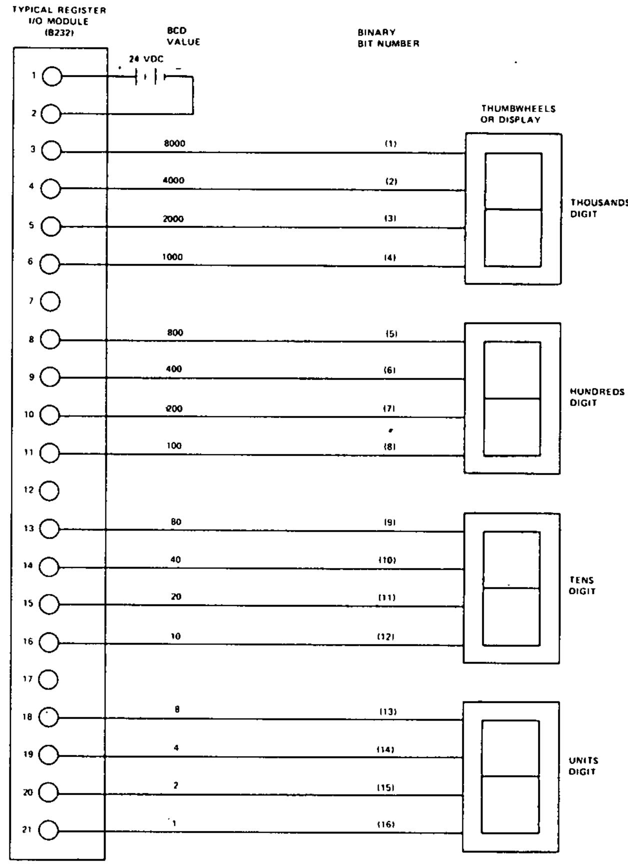


FIGURE 53 WIRING REGISTER BCD INPUT DEVICES

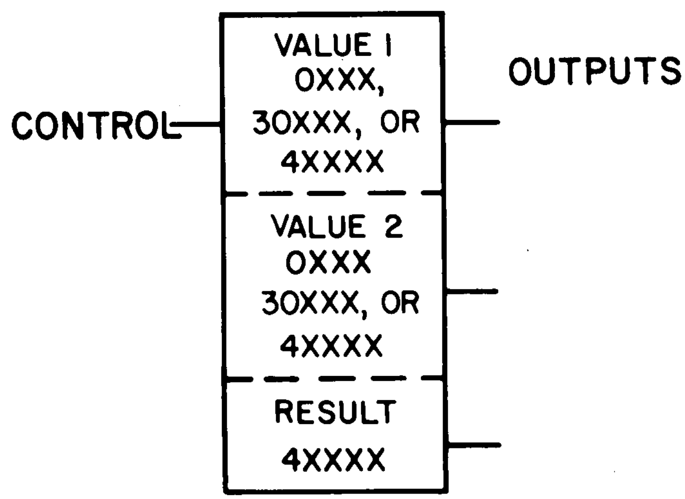


FIGURE 54 ARITHMETIC GENERAL FORM

3.4 Arithmetic Functions

All arithmetic operations require three elements placed vertically, one above the other (see Figure 54). Both the top element and the middle element can be a fixed three digit value (maximum 999) or a register reference; however, the bottom element must be a holding register (4XXXX reference).

> **NOTE:** Controllers with extended references can utilize fixed values of up to four digits (maximum 9999) in the top or middle element.

Each arithmetic function has one input (at the top left) that controls when the operation is performed and up to three output nodes (at the right). For each scan of the controller that the input node receives power flow, the arithmetic function is performed. Depending upon type, up to three outputs are possible from a single arithmetic function. These outputs are updated for every scan that the input node receives power.

NOTE: No outputs can be energized until the control input receives power flow.

484

ADDITION

This function adds the upper value to the middle value, and places the sum into the holding register addressed in the lower element. The upper or middle values can be fixed (000 to 999) or register references (30XXX or 4XXXX); the lower element must be a holding register (4XXXX). Only one output is provided with this function. If the result of an add function is a value greater than the holding register can store (9999), the 4 least significant digits are placed into the holding register. For example, if 6850 is added to 7325, the result is 14,175. The value 4175 (four least significant digits of the result) is placed in the holding register and the upper output is energized to indicate an overflow has occurred. The overflow always represents the value 10,000.

EXAMPLE - ADDITION

Referring to the example in Figure 55, when inputs I0017, 10123, and 10095 are energized, the content of register 40027 is added to the fixed value 0500, and the sum placed in register 40021. If the content of register 40027 was 5235, the result placed in register 40021 would be 5735; this operation is performed every scan that the input node receives power flow. In this example, the output is utilized and connected to coil 00033; the output's use and assigned coil are optional functions that can be adjusted by the user. With the previously assumed value in register 40027, the output would not pass power; thus coil 00033 would be OFF. If the value in register 40027 was greater than 9499 (i.e., 9563), the output would energize coil 33. In the last example, since 9563 plus 500 is 10,063 and register 40021 can contain only a four digit value, the quantity 0063 would be placed in register 40021, and coil 33 would be ON for the remainder of that scan. If the addition was performed again(i.e., inputs 10017, 10123, and 10095 are energized,) quantity 0563 (0500 plus 0063) would be placed in register 40021 and coil 33 would be OFF.

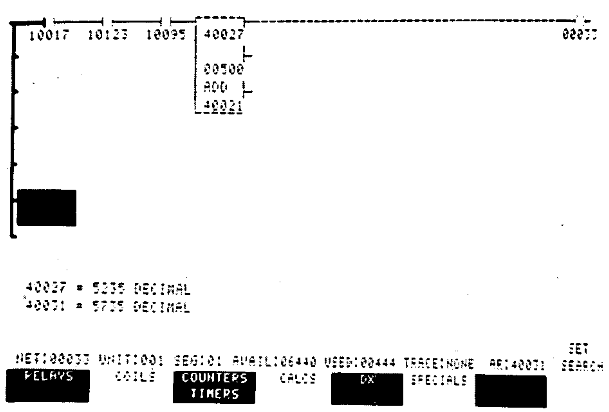


FIGURE 55 SAMPLE ADDITION

SUBTRACTION

The subtraction function operates similarly to the add function, except that the difference of the upper and middle elements is placed in the lower element holding register. Additionally, there are three outputs to the right of the numerical elements. The upper output passes power when the upper value is greater than the middle value. The middle output is energized when the upper value and the middle element are exactly equal. Finally, the lower output passes power when the upper element is less than the middle element. The use of any or all of these outputs is the user's option. Adjacent outputs can be connected with a shunt to provide greater than or equal, or less than or equal functions when these are required (i.e., set point control and alarm limits).

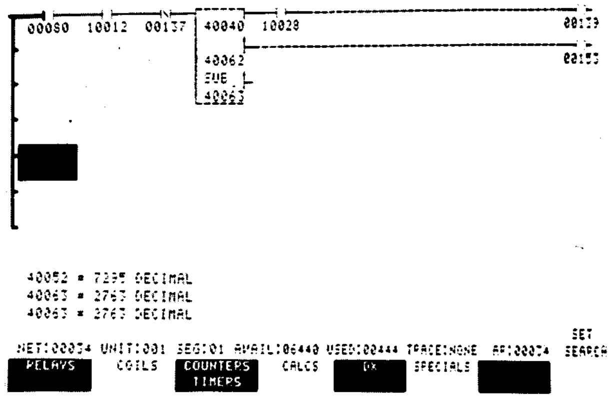


FIGURE 56 SAMPLE SUBTRACTION

EXAMPLE - SUBTRACTION

The use of the subtraction function is illustrated in Figure 56. When coil 00080 and input 10012 are energized, and coil 00137 is de-energized, the content of register 40062 is subtracted from the content of register 40040, and the difference placed in register 40063. If the content of register 40040 is 7295 and register 40062 contains 4532, when power flows to the input node, register 40063 is loaded with the value 2763. This subtraction occurs for every scan the input node receives power. When input 10028 is energized and the subtraction is performed, coil 00139 will be energized since the result is a positive value (7295 - 4532 = + 2763). If the value in register 40062 is increased equal to or greater than 7296 while register 40040 remains the same, coil 00139 would be de-energized since the subtraction function would no longer output power (7295 - 7296 = − 1). In this case, the lower output would be passing power.

NOTE: The value placed in register 40063 (in this example) is the absolute value of the difference. In other words, no sign is associated with the content of register 40063.

When the values in register 40040 and 40062 are equal, the middle output will pass power and coil 00153 will be energized.

MULTIPLY

The multiply function takes the product of the upper value and the middle value, and places the result in two consecutive registers as indicated by the lower element. The multiply is performed every scan that the input node receives power flow. One output is utilized opposite the upper element which provides power whenever the multiply is performed (i.e., control input receives power). Since the multiply of four digit values can result in an answer up to eight digits, the product is stored in two consecutive holding registers. For example 1507 multiplied by 3251 is 4,899,257. This result is separated into 0489 (high order) and 9257 (low order) for storage into two holding registers. The register addressed in the lower element always receives the high order portion of the product, even if it is zero;

485 over

the next holding register in numerical sequence receives the low order portion of the product.

NOTE: The lower element can not be programmed with the last available holding register provided with that controller's configuration.

EXAMPLE - MULTIPLY

As illustrated in Figure 57, multiply operates upon two, 4 digit numbers to produce an eight-digit product. In this example, when input 10007 is energized, the value in register 40005 is multiplied by the value in register 40059 and the resulting product stored in registers 40036 and 40037.

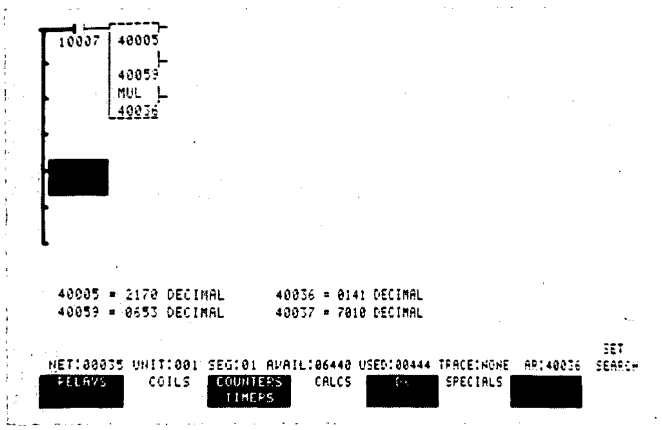


FIGURE 57 SAMPLE MULTIPLY

If registers 40005 and 40059 contained the values 2170 and 0653, the product would be 1,417,010 when input 10007 is energized. Thus, registers 40036 and 40037 would be loaded with the values 0141 and 7010 respectively. This multiply will happen for every scan input 10007 is energized. If the values in registers 40005 or 40059 are altered, the resultant product will change accordingly. In this example, the output is not utilized since its use is optional depending upon the control requirements. If an output was used, the upper output would pass power whenever input 10007 is energized.

DIVIDE

The divide function operates similarly to the multiply function, except that it calculates the quotient of two numbers. The quantity referred to in the upper element (dividend) is divided by the quantity referred to in the middle element (divisor) and the resultant quotient is placed in the holding register indicated by the lower element. The upper element is a double precision number (maximum 99,999,999) utilizing two consecutive registers (possibly the result of a multiply).

NOTE: If the upper element is a fixed value, it becomes the dividend as a single precision value. If it is a register reference, it can not be the last available register of that type (input or holding), since the upper element requires two consecutive registers.

The middle element is the divisor, either a single register reference (30XXX, 4XXXX) or a fixed value up to 9999 (24 bit machine). The lower element is a holding register reference (4XXXX) in which the quotient is stored. The remainder is also made available in the next holding register.

NOTE: To insure that sufficient room is available for both the quotient and the remainder, the holding register in the lower element can NOT be the last available register.

This remainder can be a whole number or a decimal fraction. Whichever form the remainder takes is controlled by the middle input element. The divide function is the only arithmetic function that uses two inputs. When the middle input does not receive power flow, the remainder will be a whole number. If power does flow to this input, the remainder will be a decimal fraction.

All three outputs are utilized by this function, and all require that the control input receives power to operate. The upper output will pass power whenever the division is successful; the lower two outputs indicate when division is unsuccessful. The middle output will pass power if the quotient is too large (but not infinite) to fit on one register (exceeds 9999). The lowest output will pass power if the divisor is exactly zero. When either of the two error outputs are energized, the value 0000 is placed into the lower element's registers as the quotient and the remainder.

EXAMPLE - DIVIDE

The operation of the divide function is illustrated in Figure 58. When coils 00035, 00091 and input I0046 are all energized, the content of register 40090 and 40091 (double precision number) is divided by the content of register 40130. The result is placed into 40053 and remainder into 40054. Every scan that the input receives power flow, the division is accomplished. Assuming the values in registers 40090, 40091, and 40130 are 0105, 7429, and 0136, respectively, the value 7775 (1,057,429 divided by 136) will be placed in the holding register 40053. If the second input receives power flow (as in this example), the value 2132 (the decimal remainder) is placed in register 40054 representing a quotient of 7775.2132. When no power flow occurs at the second input, the whole number remainder (0029) is placed in the register 40054. If the value in register 40130 is reduced to 125, the value placed in register 40053 becomes 8459 (1,057,429 divided by 125). If the content of register 40130 becomes zero, or register 40090 becomes 0136 or greater, the quotient becomes too large for register 40053 to store.

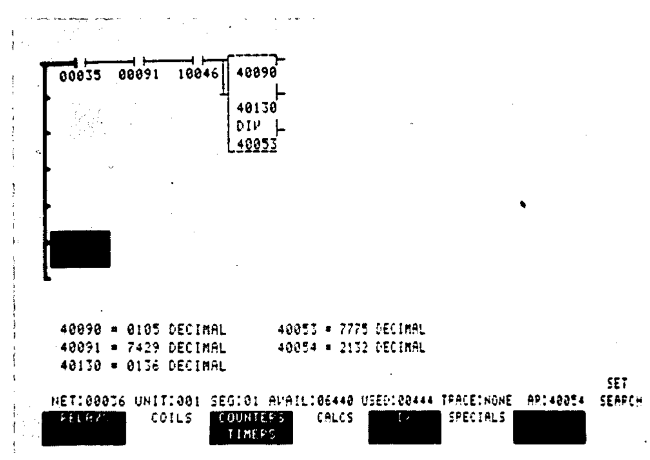


FIGURE 58 SAMPLE DIVIDE

486

ADDITIONAL EXAMPLES OF ARITHMETIC LOGIC

The following are some additional examples of arithmetic functions. These examples illustrate typical applications for this capability.

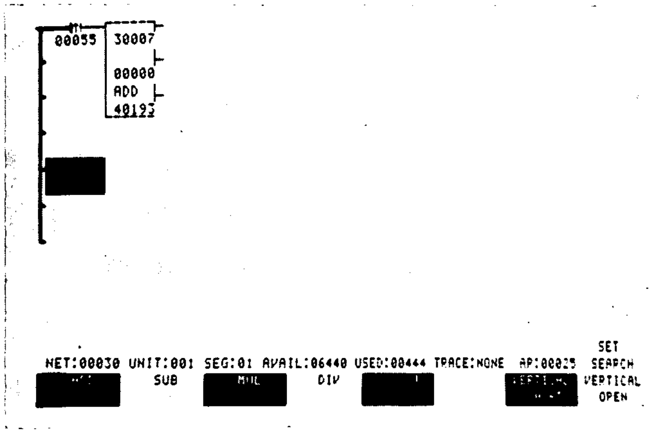


FIGURE 59 REGISTER-TO-REGISTER BCD MOVE

EXAMPLE - REGISTER-TO-REGISTER MOVE

In many situations, a numerical value must be moved from one register to another. This capability can be used to sample and hold values, alter presets on timers or counters, change pointer values, etc. As shown in Figure 59, when the control is closed (transitional contact referenced to coil 00055), zero is added to the value in input register 30007, and the result (i.e., value in 30007) is placed in register 40193. Since a transitional contact is utilized, the value in register 30007 is sampled only for one scan. This is a typical hold circuit.

NOTE: Only decimal numerical values or binary values equivalent to 9999 or less, should be used with this form of a register-to-register move.

EXAMPLE - CLEARING A REGISTER TO ZERO

Figure 60 illustrates a method for clearing a single holding register to zero upon command. When input 10072 is energized, register 40419 will be cleared to zero. For every scan that the control (input 10072) passes power, 40419 will be cleared regardless of its content.

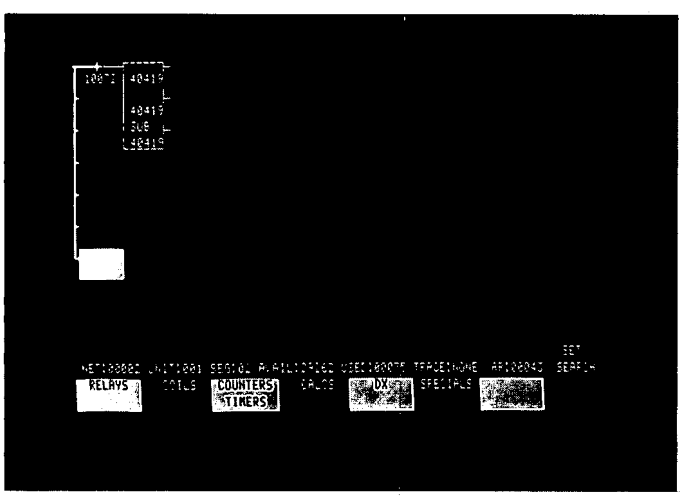


FIGURE 60 CLEARING REGISTER TO ZERO

EXAMPLE - DOUBLE PRECISION ADD

If the result of an addition is a value that can not be stored in a single register (values greater than 9999), two registers can be assigned to store the entire result. Normally two adjacent registers are assigned to store the value; however, this is not required. In the example in Figure 61, register 40116 stores the high order portion and register 40117 the low order value. If the value in 30001, when added to the current value in 40115, is greater than 9999, the upper output of the first arithmetic function passes power. This output causes the second arithmetic function to increment the value in register 40116 and, thus, record the overflow (value 10,000) in the high order register. Additional Add Logic could be included in this network which results in triple precision addition.

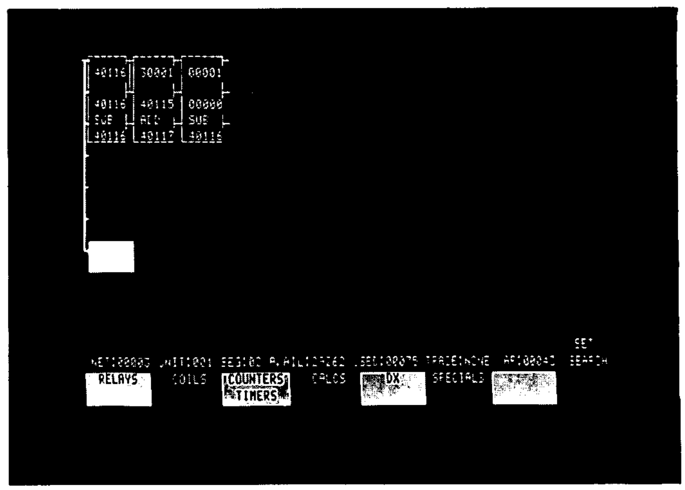


FIGURE 61 DOUBLE PRECISION ADD

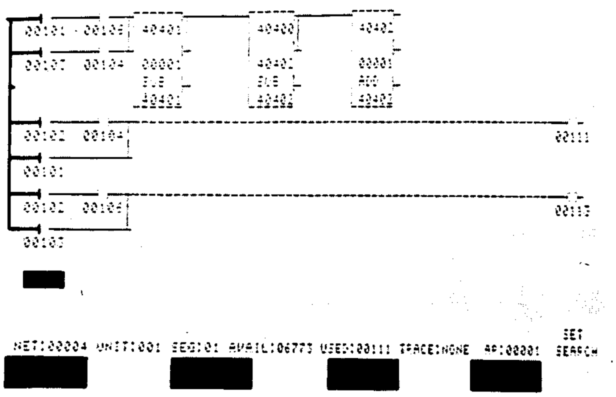
EXAMPLE - DOUBLE PRECISION SUBTRACTION

The double precision subtraction is slightly more involved than the addition, but also provides a comparison function as illustrated in Figure 62. The two numbers for illustrative purposes are assumed to be in registers 40473 and 40474 (HO, LO of first number respectively) and registers 40327 and 40328 (second number). The double precision result of the subtraction (number 1 minus number 2) is placed in registers 40401 and 40402. Coil 00111 indicates a positive result (number 1 greater than number 2), coil 00112 indicates a zero result (equal magnitudes), and coil 00113 indicates negative result.

The two subtraction functions of the first network perform the double precision subtractions (HO and LO), as well as determine the relative magnitudes of each value. A carry of 10,000 from HO to LO is required only if the result of each individual subtract results in an opposite sign (i.e., HO positive and LO negative or HO negative and LO positive). The arithmetic functions of the second network perform the carry when required; register 40400 has the fixed content 9999.

To validate this example, try nine sets of numerical values, combining all possible combinations of magnitude relationships for HO and LO portions (greater than, equal, and less than for each portion). The double precision subtraction with compare requires 56 words (including numerical storage); if the compare function is not required, the logic driving coils 00111, 00112, and 00113 can be deleted, reducing memory requirements to 53 words.

487 over



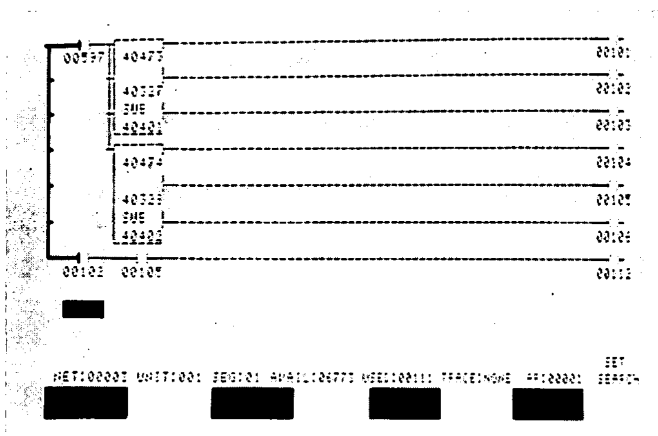


FIGURE 62 DOUBLE PRECISION SUBTRACT

EXAMPLE - CONVERSION OF ANALOG SIGNAL TO ENGINEERING UNITS

Analog inputs utilizing the 200 Series I/O are 12 bit conversions, resulting in magnitudes from zero to 4095. In many applications, these analog signals are provided to operators in units (pounds, gallons per second, degrees C, feet per minute, etc.) via LED displays, CRT monitors, or report printouts. The multiply function can perform this conversion with ease. To illustrate the technique, assume a 4 to 20 ma signal represents pressure from 50 psi to 400 psi (or temperature from 50 degrees F to 400 degrees F or another analog measurement).

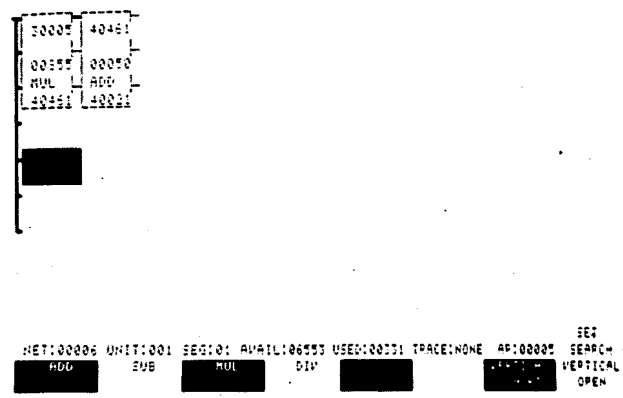


FIGURE 63 ANALOG CONVERSION TO ENGINEERING UNITS

Figure 63 shows the logic to generate the display in register 40021 from an analog signal in register 30005. The range of the analog signal in units (e.g., 400 psi -50 psi = 350 psi) is divided by 4095, the internal range of the analog signal. The resultant quotient when multiplied by 10,000 (e.g., 855) is used as the multiplier. Since the minimum analog signal represents a non-zero value (e.g., 50 psi), the addition block corrects the display. Note that only the high order result of the multiply is used and that no compensation for roundoff is used in this example.

488

Section V — Internal Principles

5.0 Theory of Operation

5.1 Introduction

The 584 Controller is the culmination of years of MODICON's experience with industrial programmable controllers. It provides speed, flexibility, size, and sophistication unmatched by its competitors. The controller can be effectively applied to small relay replacement applications (replacing 50-100 relays), sophisticated data collection, and report generation applications that previously required mini-computer support. The programming format (up to seven rungs with up to ten elements in any configuration) provides maximum flexibility as well as minimum memory utilization. Elements can be programmed anywhere in this format and use memory only where actually entered.

This section describes the system software data base detailing memory utilization and conventions designed by MODICON for proper system operation. Also provided is a description of the procedures required to properly integrate the controller into a data processor system.

This information is NOT required for the designer to program the controller. The designer simply enters his program by means of the various programming devices in the multi-node (10x7) format. However, when the controller is be interfaced to a computer system, this information will assist the user in obtaining appropriate data.

5.1.1 Scan

The MODICON 584 Controller processes its logic by solving networks consecutively by number, beginning at network 1 and continuing through the last network programmed into the controller.

NOTE: Controller scans only memory actually programmed with logic.

A network is defined as a group of connected programmed elements; these elements can be relay contacts or numerical references. During logic solving, individual input and output modules are serviced once each scan. As soon as one scan is completed, the next scan commences with network 1.

Each network is independently solved from column one through column ten. Within each column the logic is solved from the top rung to the bottom rung of that column. The new results from each network (either coil status or register content) are immediately available for use by the next network or column. The scan is by network, not by coil number. This scanning technique is basic to the operation of the 584 Controller.

Upon each scan every network is solved, and each I/O module is serviced. The exact time to complete a scan varies from application to application, but it basically depends on the quantity of I/O modules to be serviced. The time to complete a scan is usually not impacted by network logic solving unless more than 4K of memory is used in any segment. Table 13 provides some sample scan times for typical applications.

NOTE: When power is applied, a power-up sequence is performed. The time required for power up operation varies depending upon memory size.

TABLE 13
SAMPLE SCAN TIMES - TYPICAL 584 APPLICATIONS

	Memory Configuration			I/O Quantity (Points)		Equivalent Typical	
Size	Elements	Registers	Input	Output	Totals	Relays	Scan Time
4K	3,450	250	240	160	400	350	8mSec
8K	7,200	450	496	336	832	720	15mSec
8K	6,300	1200	640	320	960	630	20mSec
12K	11,000	650	752	496	1248	1100	25mSec
12K	10,000	1900	864	384	1248	1000	25mSec
16K	14,600	850	1024	656	1680	1500	30mSec
16K	8,000*	2800	1344	528	1872	900	35mSec
16K	7,000*	4000	512	320	832	700	15mSec
24K	22,200*	1250	1360	912	2272	2100	40mSec
24K	11,500*	5000	1920	736	2656	1500	50mSec
24K	9,500*	8000	1616	640	2256	1000	40mSec
24K	9,000*	9000	768	480	1248	900	25mSec
32K	20,000*	1200	1872	1104	2272	2200	40mSec
32K	15,000*	8000	2048	992	3040	1600	55mSec
32K	13,500*	9999	1024	640	1664	1500	30mSec

NOTES: *Indicates configuration using extended references. Quantity of elements do NOT include allowances for over head. I/O Quantities are a total of discretes and register I/O.

489 over→

After this power up sequence, scanning is performed based upon read data (inputs, disables, and latched coils updated) beginning with network 1. If a power failure is detected, scanning is terminated at whichever network is currently being solved, and the power-down sequence is initiated which includes turning all outputs OFF.

The logic is solved first by segments and then by networks within a segment. Between segment solutions there is an exchange of I/O information as illustrated in Figure 82. Up to two channels of inputs (256 points) and two channels of outputs (256 points) can be exchanged after each segment. The total quantity of I/O channels and the number of segments in any 584 Controller are system parameters that can be changed by the MODICON Service Center or by the P190 Programmer with the Utility tape. The amount of networks (and, thus, quantity of memory) allotted to each segment is controlled by the user as he enters the program. Prior to starting a new network, the user must indicate which segment the network is to be placed. New networks are normally added at the end of a segment. However, they can also be added at the beginning of the segment or between any two networks within that segment. Of course, networks can be deleted in part or in total at any time.

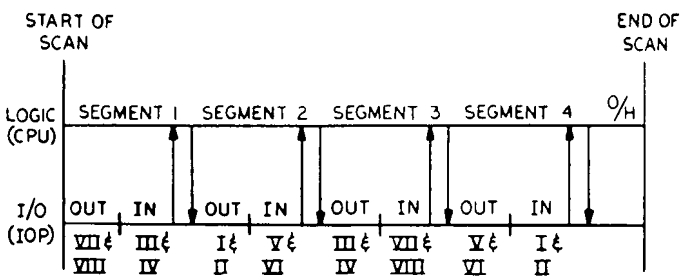


FIGURE 82 SAMPLE SCAN WITH I/O

The controller uses a dual processor technique, wherein the logic is solved at the same time the communications to the I/O modules is performed. At specific points in the scan (i.e. at the end of each segment), logic solving is briefly halted, input data is brought into the main CPU and output data is provided to the I/O processor (IOP). Thus, while any segment is being solved, the IOP is providing status to two channels of outputs (256 points) and receiving a similar quantity of inputs. Complete error checking is performed by the IOP to insure accurate information transfer to and from I/O modules. An example of an eight channel I/O system is presented in Figure 82. The channel assignments are not made in numerical relationship to segment solving. Channels I and II are read for inputs prior to segment 1 and are provided outputs after segment 1; thereafter, I/O is normally serviced by channel number as shown.

At the end of each scan various internal functions are performed; this is typically referred to as "housekeeping". Most important is to service the two MODBUS communication ports. During this time, programming changes can be made, data provided to refresh the P190 CRT display, information sent to a computer, or register content loaded from an external device. Each port is handled separately at the end of the scan so that data can be sent to both MODBUS devices upon each scan. Along with MODBUS servicing, the Register Access Panel (RAP) on the front of the mainframe is updated and responded to. Finally, if Memory Protect is ON, a portion of the retentive memory is examined with a checksum function to insure that this memory has not changed. User logic memory and system parameters (including disable status) are subject to the checksum verification. If a fault is located in this protected memory, the controller will be stopped with outputs turned OFF, and an error code will be displayed.

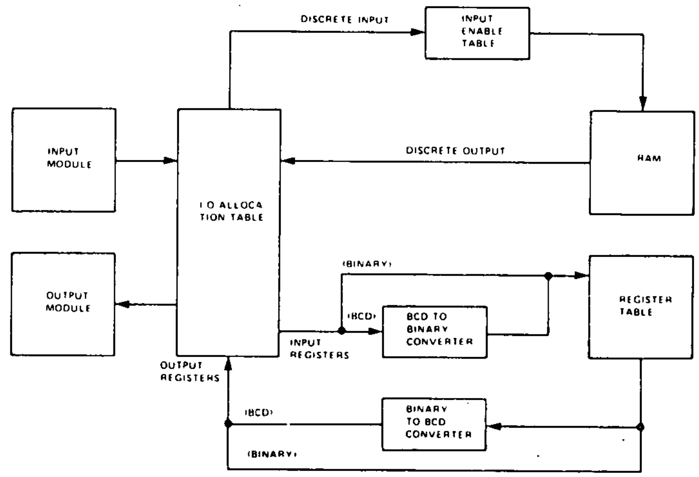


FIGURE 83 I/O ERROR CHECKING BLOCK DIAGRAM

Standard I/O error checking is illustrated in Figure 83. When servicing input modules, the processor requests the status of its 16 circuits twice and then compares the two samples. If they agree, the data is stored in the I/O status; if they do not agree, another complete sample is requested and compared to the previously obtained status. This sampling is continued until two consecutive samples agree or until five compares are made. If, after five compares, no two consecutive samples agree, the processor assumes all inputs are OFF (zero) for that input module and continues to scan. At the input module, if communications from the processor are not received within 250 ms, the module will turn its active light OFF.

Output modules are provided with new status (all 16 outputs) at least twice each servicing; both transmissions are echoed to the Processor by the output module. The output module compares both sets of received data and, if they agree, uses them to drive its outputs. If they do not agree, the Processor retransmits the data and a new compare is accomplished by the module between the most-recently received previous data and the new transmission; the re-transmission is initiated by the Processor's compare of the echoed data which will also be faulty. Up to four re-transmissions are accomplished if the echoes do not agree with the transmitted data. If, after a total of five comparisons, a valid compare is not obtained, the outputs retain their previous states and the Processor continues scanning. If the output module does not receive valid data within 250 ms, it will turn its active light OFF and shut all outputs OFF.

Once an I/O module has been detected as malfunctioning (i.e., requires six transmissions), it is logged internally. This data is made available to the user's logic via the MOVE STATUS function (see paragraph 4.1.7). On subsequent scans, only two transmissions are made to the I/O module. If a valid response is made to these two transmissions, the log is corrected, and the CPU will, if necessary send up to six transmissions on the next scan. If an invalid response is obtained, no further transmissions are made on that scan, and the log continues to reflect a faulty module. Note that error checking is accomplished individually on all 16 bits obtained or provided to the I/O module. It is not a parity or error code verification.

5.1.2 Memory Utilization

The logic entered by the user is stored in memory in successive words. This logic is stored by network number, with the first network at the beginning of user logic memory (first network to be scanned). If new networks are to be added, they are normally placed after all existing networks. However, the P190 Programmer has the ability to either add logic elements to an existing network or to insert networks between existing networks. If additional memory is required, existing logic is moved down as necessary to make room for the new logic.

NOTE: The following discussion assumes one word per element. If the 584 Controller utilizes the extended reference option, all calculations should be increased to one and one half word per element.

Each network is stored in memory by vertical columns. There is one element at the beginning of each column that indicates the placement of elements and vertical connections within that column. The entire first column is stored, then the second column, etc. Coils are stored and solved in whatever column they are programmed, although the P190 Programmer will display the coil adjacent to the right leg of the ladder diagram. However, since coils are optional, there is also one element which indicates the start of a network. This is the first element in memory for each network. The first element stored in the user logic memory is network one's start of network element.

Using a word of memory for each column allows contacts and other programmed elements to be placed anywhere in that column and, thus, anywhere in the network. There are no restrictions on placing any logic element except that coils must be the last element in a horizontal row. The P190 Programmer displays these coils at the right side of the row to indicate that no logic space is available to the right of the coil. Of course, other rows (all of them) can extend beyond any coil and are terminated only by their own coils. Within a column, locations where no elements are desired can be left vacant without expending logic memory beyond the single start of column word. Also, vertical connections can be placed anywhere (as many as necessary) without using additional memory. Vertical connections do not have to be continuous. However, if horizontal shunts are desired, they are considered a programmed element and will use one word of memory for each one programmed.

As an example refer to the logic shown in Figure 84. This network utilizes four horizontal rungs (W,X,Y, and Z) and nine vertical columns (1-9). To determine the exact quantity of memory words required for this network, each column is analyzed as follows:

Column 1: Only the A and B elements require memory. The enclosed space opposite rung X, as well as the vacancy opposite rung Z do not require memory. Adding the beginning of column element results in 3 words for this column.

Column 2: Similar to column 1, only elements C and D require memory, total 3 words.

Column 3: The E element and the horizontal shunt opposite the Y rung each require one word, total 3 words. Note that the vertical connections to the right of element E do not require additional memory since they are included in the start of column element.

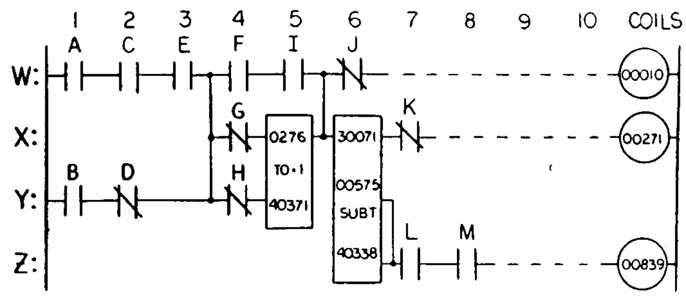


FIGURE 84 SAMPLE LOGIC FOR MEMORY UTILIZATION

Column 4: All three elements (F,G, and H) require one word of memory total 4 words of memory.

Column 5: All three elements (contact I, preset 0276, and register reference 40371) require one word of memory. The vertical to the right of contact I does not require any additional memory; total 4 words of memory.

Column 6: All four elements require one word of memory each. The vertical connection to the right of the Subtract block does not require memory; it is specified by the start of column element. Total requirements are 5 words of memory.

Column 7: One word of memory is required for each element: coil 00010, contact K, and contact L, a total of 4 words of memory.

Column 8: There is no requirement for memory in the top (W) rung. Only coil 00271 and contact M require memory-total 3 words of memory.

Column 9: Only coil 00835 requires memory, total two words.

Summary of required memory is as follows:

Column	Words
1	3
2	3
3	3
4	4
5	4
6	5
7	4
8	3
9	2
Start of Network	1
TOTAL	32

NOTE: On controllers configured for extended references, a total of 48 words (each 16 bits) is required for this network.

Although the coils are displayed by the P190 Programmer in the extreme right column, they are actually placed in memory after all the programmed logic in that horizontal row. Within the user's memory, coil 00010 is placed before contact K as the first element in the 7th column. Similarly, coil 000271 is placed before contact L and coil 00839 after contact M. Elements programmed in columns 6-9 are stored in the following order: start 6, contact J, register reference 30071, fixed value 00575, register reference 40338; Start 7, coil 00010, contact K, contact L; Start 8, coil 00271, contact M; Start 9 and coil 00839. Thus, coil 00010 will be solved before coil 00271, which in turn is solved before coil 00839 since the logic is solved by columns.

491
over

5.1.3 Data Base

Details of the data base are NOT required to effectively program the 584 Controller. The following information is provided to document the internal operation of the controller and, thus, complete the user's understanding of this controller. User's who would be interfacing the controller to a computer do not require knowledge of the internal operation of the controller since the MODBUS protocal uses reference numbers and not internal memory locations.

In order to obtain the high speeds within the 584 Controller, there are actually two alterable memories in its internal architecture as illustrated in Figure 85. The retentive memory is that which is varied to create the different 584 sizes (4K, 8K, 12K, 16k, 24k, or 32K); the volatile memory has two sizes which allow two reference ranges basic (2K each) or expanded (10K each). The volatile memory operates at very high speed and contains the ON/OFF status of discrete and content of registers; however, it is not retentive upon power failure. If AC power should be lost, the processor stops solving logic and stores the current state of the RAM in the retentive memory. Upon restoration of AC power, the data is then moved from the retentive memory into the RAM prior to solving any logic.

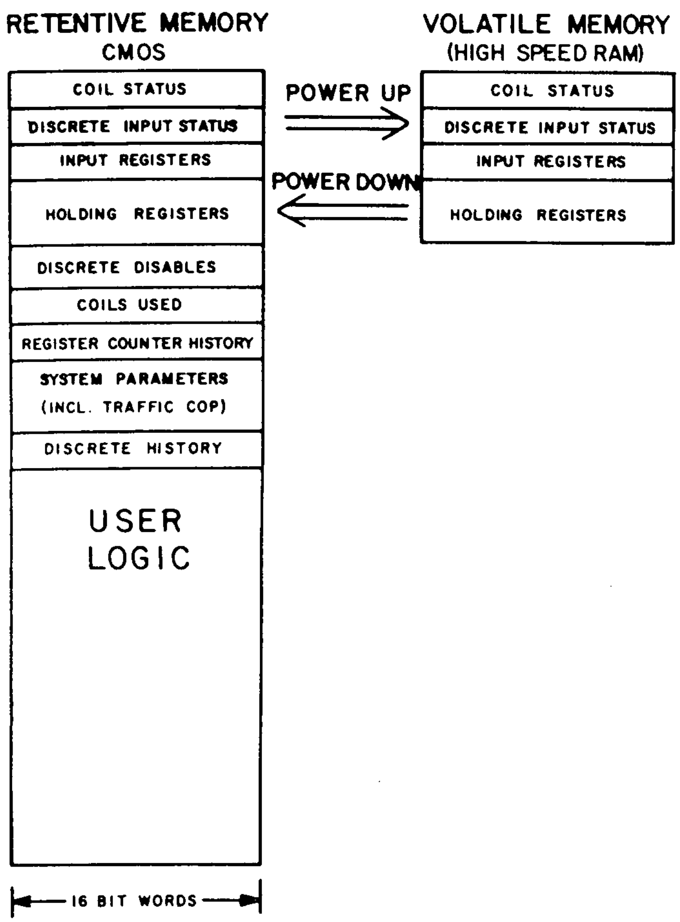
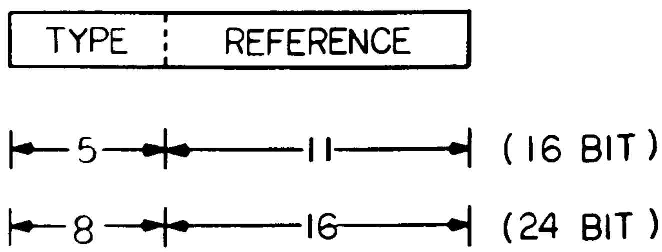


FIGURE 85 INTERNAL MEMORY BLOCK DIAGRAM

As the retentive memory is configured by the P190 Programmer using the 584 Utility tape, the amount of memory available for the user's logic is displayed. Discrete references are stored sixteen per word; they must be configured in groups of sixteen. In addition to space for ON/OFF status, memory is also allocated to disable conditions, history (previous scan's state), and used coil data. Registers require one word for each and can be allocated in any quantity desired. A double status for each holding register is also required to store the counter history, one if the register stores the result of an up counter and the other for a down counter. Each controller is allocated a relatively small quantity of words (256) for system parameters, such as quantity of segments and configuration of memory.

LOGIC STORAGE

Within the logic area of retentive memory, the logic is stored by networks, columns within the networks, and finally by elements within the column. The element becomes the basic building block of the user's logic. Each element utilizes one 16 bit word (basic references) or one and one half words total 24 bits (extended references). This section describes in detail the 16 bit format and provides sufficient information to understand the 24 bit format. In general each element is created in two portions as follows:



The first 5 bits allow 32 specific element types, and the remaining 11 bits of the 16 bit format allow 2048 references. The 24 bit format expands these values to 256 element types (8 bits) and 65,536 references (16 bits). Currently, the capacity of the 584 Controller only requires 5 bits for element types since less than 32 element types are used even with all its capabilities. The 24 bit data base is required only when additional references are necessary; sufficient RAM memory in the largest model is provided to support a maximum of 9999 holding registers and 8192 discrete references. The internal logic solving operates on a 24 bit data base. To convert from 16 bit to 24 bit operation, zeros are filled in to the left or high order of each portion of the 16 bit element.

```
               ELEMENT            REFERENCE
                                        3 2 1
16 BIT:        1 1 1 0 X    Y Y Y Y Y Y Y Y Z Z Z

           3 2 1                                3 2 1
24 BIT:    Z Z Z 1 1 1 0 X   Y Y Y Y Y Y Y Y Y Y Y Y Y Z Z Z
```

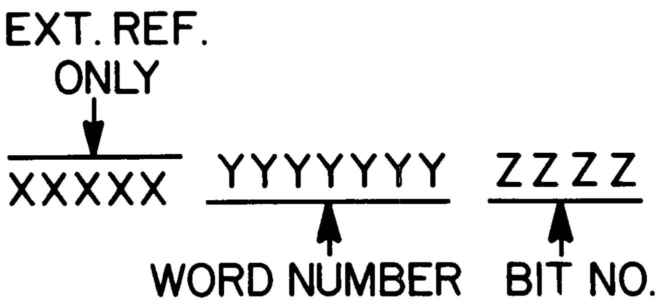
WHERE: X=0 FOR MOVE, 1 FOR MATRIX
Y=TABLE/MATRIX LENGTH IN BINARY
Z=SPECIFIC OPERATION TYPE

492

TABLE 14
SUMMARY OF ELEMENT TYPES

Binary	Decimal	Definition	Reference
00000	0	Beginning of Column	Special
00001	1	Beginning of Column	Special
00010	2	Beginning of Column	Special
00011	3	Beginning of Column	Special
00100	4	Start of Network	None
00101	5	I/O Exchange/End of Logic	Type Selection
00110	6	Null Element	None
00111	7	Horizontal Shunt	None
01000	8	Normally Open Contact	Discrete
01001	9	Normally Closed Contact	Discrete
01010	10	Positive Transitional Contact	Discrete
01011	11	Negative Transitional Contact	Discrete
01100	12	Coil-Non-Retentive	Coil
01101	13	Coil-Retentive (Latch)	Coil
01110	14	Skip Constant Quantity	Fixed Value
01111	15	Skip Register Quantity	Register
10000	16	Constant Value Storage	Fixed Value
10001	17	Register Reference	Register
10010	18	Discrete Group Reference	Discrete
10011	19	Down Counter	Holding Register
10100	20	Up Counter	Holding Register
10101	21	Timer Seconds	Holding Register
10110	22	Timer 1/10 Second	Holding Register
10111	23	Timer 1/100 Second	Holding Register
11000	24	Add	Holding Register
11001	25	Subtract	Holding Register
11010	26	Multiply	Holding Register
11011	27	Divide	Holding Register
11100	28	Move Function	Fixed Value
11101	29	Matrix Function	Fixed Value
11110	30	Spare-Future Use	Fixed Value
11111	31	Spare-Future Use	————

Table 14 summarizes the five bit coding for element type and the values allowed in the reference area (11 bits or 16 bits depending upon format). Most of these elements require little or no explanation and are self-evident; however, a brief discussion of each type or group of elements will be presented for the sake of completeness. The beginning of a column is only one element type, yet it requires four element positions. Since this element includes the position of vertical connections between adjacent rungs (6 bits) and the position of up to seven elements (7 bits), there is insufficient room in the reference area for 13 bits of data. Thus, two bits are borrowed from the element type; the remaining three bits (000) are unique relative to other element types and specify a beginning of column element. The overall format of this element is as follows:

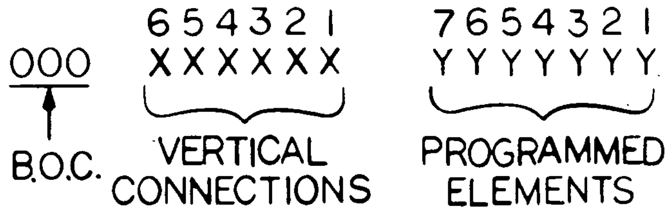


Vertical connections are specified by the X bits with X1 a one if there is a connection between rungs 1 and 2, X2 a one for connections between rungs 2 and 3, etc. The Y bits indicate location of programmed elements, Y1 a one if there is an element programmed in the top or first rung, Y2 indicates the presence or lack of an element on the second rung, etc.

The start of network, null element, and horizontal shunt have no reference associated with them, thus, the reference area is ignored and normally filled in with zeros. The I/O exchange/end of logic element performs dual functions, and the reference area indicates which function a specific element is performing. If the reference area is all zeros, the element indicates both an I/O exchange and the end of logic storage. As discussed in section 5.1.2, specific housekeeping is performed at the end of the logic scan. A reference area of binary value one indicates only that the I/O is to be exchanged and the logic scan should continue.

The relay contact types all require a reference. This reference can be to any logic coil or discrete input that is valid with the current memory configuration. The reference area is divided into three areas with X bits valid only with expanded references.

493 over



As indicated in Figure 86, coils are placed in the top or low address words of the high speed RAM followed by the discrete inputs. How many words are assigned to coil status and how many to discrete input status depends upon the configuration. If a configuration allowed 736 coils and 1312 discrete inputs, there would be 46 words assigned to coils and 82 words assigned to discrete inputs. These words total to 128 which can be addressed (starting at zero) solely by the Y bits; thus, up to 2048 total discrete references can be accumulated with the basic single word reference. If more total discrete references are desired, the X bits must be used which allows up to 4096 word references or a total of 65,536 discretes!

NOTE: There is no 584 model that currently offers RAM space for more than 8,192 discrete references.

The Z bits refer to the individual bit in the RAM where the status of that reference is obtained. The low order bits starting with bit zero contain the status of the low magnitude reference numbers. For example, coil 00001 status is stored in word zero, bit zero (right hand end), coil 00015 is word zero bit 15 (left hand end), coil 00037 is word four bit 2, etc. The input reference depends upon the quantity of coils in that configuration. For example, if a controller were configured for 736 coils and 1312 inputs, input 10001 would be word 46 bit zero. To obtain the stored relative reference, subtract one from the reference coil number and divide the result by 16; the quotient is the word number and the remainder is the bit location in that word. For inputs, subtract 10,001 from the reference number, divide by 16 and add to the quotient the number of words assigned to coil references. The result will be the word address and the remainder the bit address.

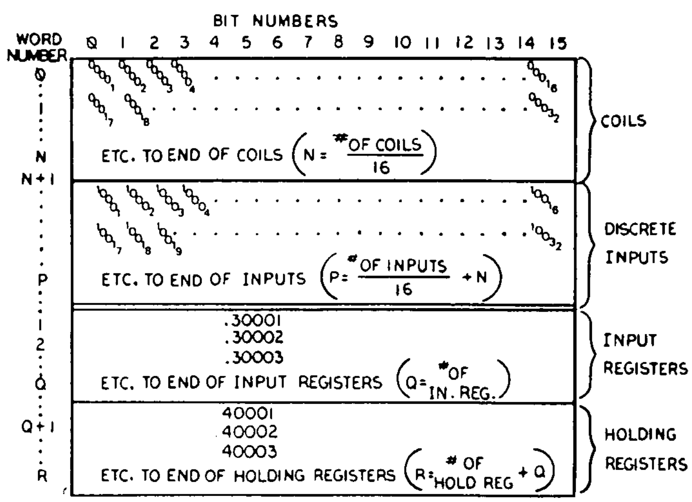


FIGURE 86 DETAILS OF HIGH SPEED RAM

There is a word reference that indicates relative position in the RAM starting at zero and a bit reference to indicate which bit is to be changed. The coil element changes the state of the referenced bit; the relay contacts only use this state as a reference, they can not change it.

Beyond the housekeeping and relay elements, there are a group of non-relay elements such as timers, counters, and functional blocks. There is also a group of three elements that are used with all non-relay elements; these are constant references, register references, and discrete group references. The following is a discussion of the three references that should be understood before proceeding to the non-relay elements. The constant value is a quantity that is fixed at time of programming and can not be dynamically altered by the controller's logic. The reference area stores the binary equivalent of this value up to a maximum magnitude of 999. If a controller is configured for extended references, additional space is available in the reference area (11 vs 16 bits) that allows constant values up to 9999 to be stored. Although larger values could be stored in binary representation, the P190 and the architecture of the 584 Controller limit the maximum to be a convenient decimal value.

The register references indicate the relative internal RAM word in which the register content is stored. Registers are stored first by input registers then holding registers; input register 30001 has a relative register address of one. Thus, the relative address of any input register is that reference minus 30000. The relative address of holding registers depends upon the quantity of the input registers. For example, if a controller were configured for 64 input registers, 40001 would have the relative address 65. To obtain the address of any holding register, subtract 40000 from its reference and add to the result the total quantity of input registers. Relative register address zero does exist when it is used during the programming of non-relay functions to indicate that the reference has not been entered. Any function with a relative register address of zero will not operate. Referring to Figure 86, registers are stored after discretes. The controller records the dividing point in RAM between discretes and registers and will automatically add that word location to the relative address stored with the element. Reference to registers need not consider the total quantity of discrete references.

The third element allows direct reference to discrete states in groups of sixteen. Recall that the discrete references (coils - 0XXXX and inputs - 1XXXX) with relay contacts are stored in two portions: a relative word address and a bit number. This element uses only the relative word portion of the discrete reference since discretes will be referenced in complete groups of 16. For example, if a controller is configured for 736 coils and 1312 discrete inputs and a reference is desired for the sixteen coils 00145-00160, the relative address (0009) starting at zero can be used. Reference to discrete inputs 10081-10096 in this controller would be (0051). To obtain relative address for coils, divide the lowest reference (e.g. 00145) by 16; note that the remainder of this division must be exactly one to be an acceptable group reference. To obtain the relative address of discrete inputs, subtract 10,000 from the lowest reference, divide the result by 16, and add the quantity of words assigned to coils to the quotient. Again the remainder of this division must be exactly one. Several of the functions will allow reference to multiple discrete groups (e.g. 2,3,4,... groups) thereby allowing large discrete references, e.g. total 32, 48, 64... (ON/OFF statuses) if they are in consecutive order.

The first non-relay element listed in Table 14 is the skip function. The constant reference is similar to the constant element previously discussed in that it is a binary value with a maximum

magnitude of 999. If a controller is configured for extended reference, this value can be up to 9999. The register reference skip element is also similar to the register reference element previously discussed. In this reference area is the binary relative reference to the register, input registers in the low values followed by the holding registers. Input register 30001 is relative address one; the exact address for holding registers will depend upon the quantity of input registers.

The Timer/Counter group of elements represent the lower element of the two element programming format. The upper element can be either a constant value element (16) or a register reference element (17) which have been previously discussed. The lower element type selected will specify the exact type of counter (Down or Up) and type of timer (seconds, tenths of second, or hundredths of second), as well as the holding register into which the current value is stored. The reference area of each of these five element types is the relative address in RAM of the holding register into which the current value is to be stored. Relative register addresses are established by both the quantity of input registers provided with that controller and the holding register reference. For the Timer/Counter Group it must be a holding register.

The calculate group of elements are the bottom element in the three element programming format. The top and middle elements can be a constant element (16) or register reference (17) as previously discussed. The lower element type selected will specify the exact type of calculate (add, subtract, multiply, or divide) as well as the holding register into which the result is stored. The reference area of each of these four element types is the relative address in RAM of the holding register into which the result is to be placed. Relative register addresses are established by both the register reference number and the quantity of input registers provided with that controller. For the calculate group, it must be a holding register. If the calculate function is a multiply or divide, the register associated with that type element must not be the last holding register available in the controller since two registers are actually used (double precision product or quotient and remainder). If the calculate function is divide, and the upper element is a register reference (type 17), it can not be the last holding register or the last input register since the divide uses a double precision dividend.

The last two element types are used with Functional Block programming. The function area (MOVE or MATRIX) is selected by the element type (28 or 29); the specific type of operation (i.e. Block Move, Search, AND, Compare, etc.) is specified in the three last significant bits of the reference area. The remaining bits in the reference indicate the length of the table/matrix up to 255 (8 bits left after 3 for operation type). Controllers with expanded references have additional bits for length (up to 13) and can utilize table/matrix sizes up to 9999. For error checking, the three bits defining operation type are duplicated at the high order end of the 24 bit expanded references.

A summary of the bit details for these two element types is as follows. The Move and Matrix elements are the bottom elements of the three element programming format. The top and middle elements can be constant values, register references, or discrete group references depending upon the specific operation selected. Table 15 summarizes the significance of the Z bits used in the above format and the permissable references (element types 16-18) for the top and middle elements. To conserve scan time, many of the functions limit the size of tables and matrices to 100. Matrix sizes are limited to 600 since a four digit pointer can not address more than 9999 bits which indicates 624 sixteen bit words. Maximum size of tables and matrices are indicated in Tables II and 12 respectively.

TABLE 15
SUMMARY OF Z BIT CODING

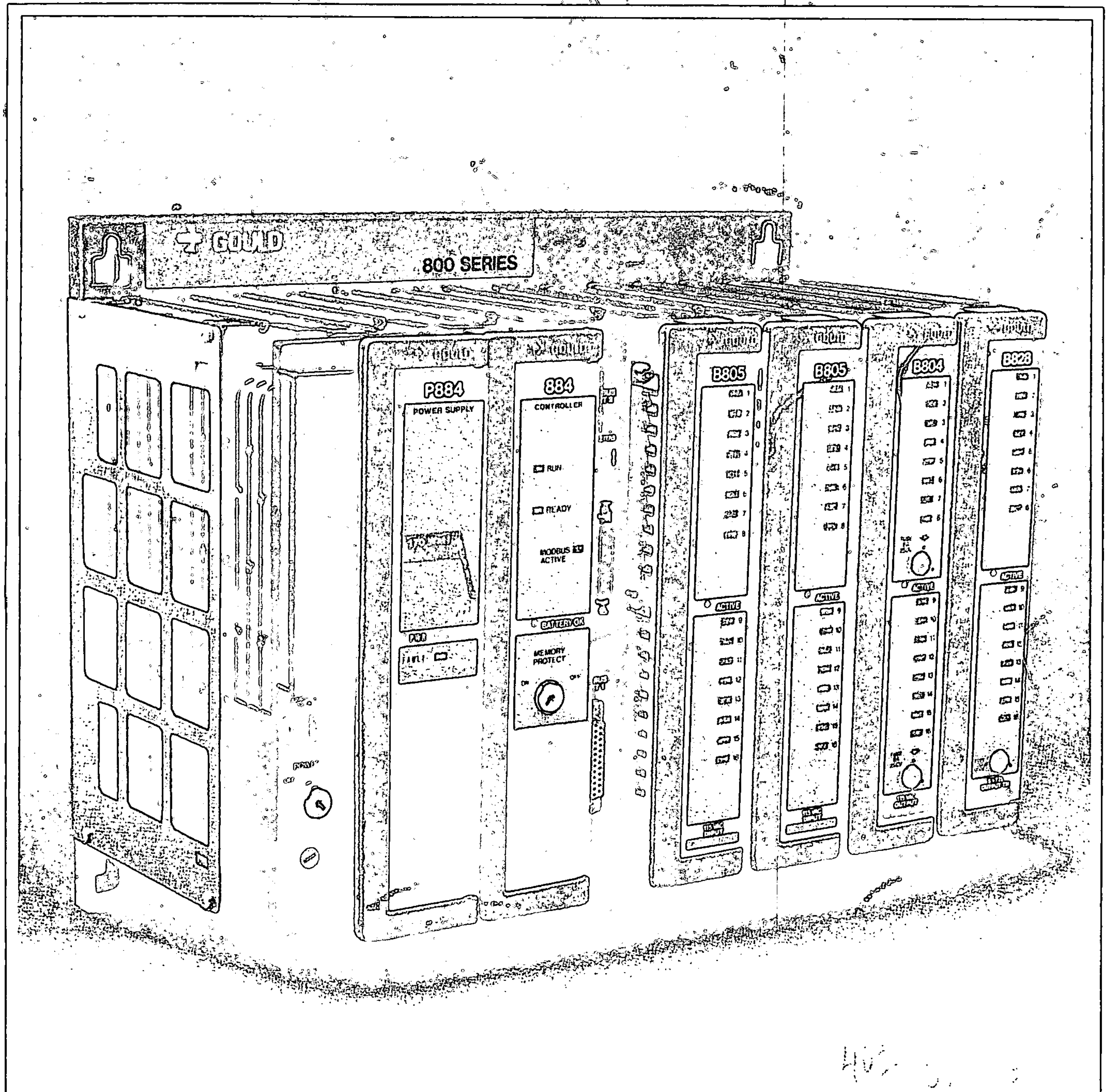
321 ZZZ	Move Functions Operation	Move Functions Top Element	Move Functions Middle Element	Matrix Functions Operation	Matrix Functions Top Element	Matrix Functions Middle Element
000	R-T	17 or 18	4XXXX	AND	17 or 18	0XXXX 4XXXX
001	T-R	17 or 18	4XXXX	OR	17 or 18	0XXXX 4XXXX
010	T-T	17 or 18	4XXXX	Compare	17 or 18	4XXXX
011	Block	17 or 18	0XXXX 4XXXX	Sense	16 or 17	17 or 18
100	FIFO In	17 or 18	4XXXX	CLR/SET	16 or 17	0XXXX 4XXXX
101	FIFO Out	4XXXX	0XXXX 4XXXX	Complement	17 or 18	0XXXX 4XXXX
110	Search	17	4XXXX	XOR	17 or 18	0XXXX 4XXXX
111	Status	0XXXX 4XXXX	——	Rotate	17 or 18	0XXXX 4XXXX

495 over →

EXHIBIT E

Gould 884 Programmable Controller

Product Description



GOULD
Electronics

Table of Contents

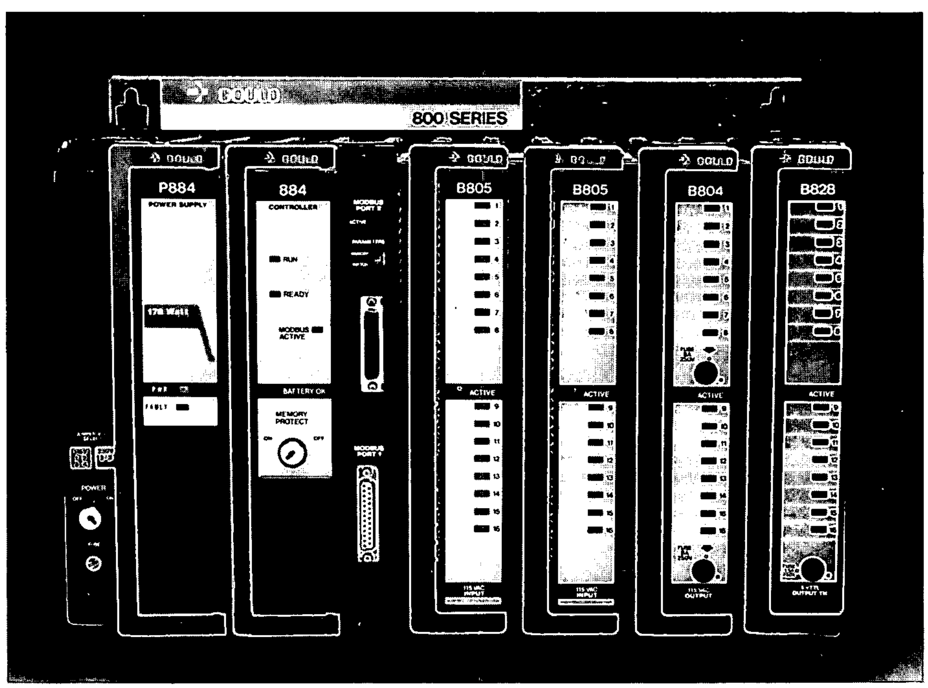
I. 884 System

Gould's 884A is a medium-sized Programmable Controller designed and built to meet present and future control needs. The 884A represents years of experience and incorporates durability, flexibility and reduced operating costs. Modular design greatly simplifies system maintenance and checkout while sophisticated functions such as Signed Double Precision Math, Real Time Clock, and Dual Modbus communication ports expand the traditional role of Programmable Controllers.

Designed with state of the art solid state circuitry, the 884A operates in most hostile industrial environments where heat, vibration, and electrical noise become critical factors. Advanced, built-in diagnostics and integrity checks insure proper operations and simplify fault-finding procedures. There is no need to learn any high-level computer languages because the 884A can be programmed using standard industrial Relay Ladder Logic.

Using the 884A Processor as a basic relay replacement system results in substantial cost savings over conventional relays in the form of wiring, panel, and start-up costs. The 884A, however, also offers increased sophistication with advanced features such as ASCII Read/Write, Real-Time Clock, Annotated Ladder Lister, and Program Merge.

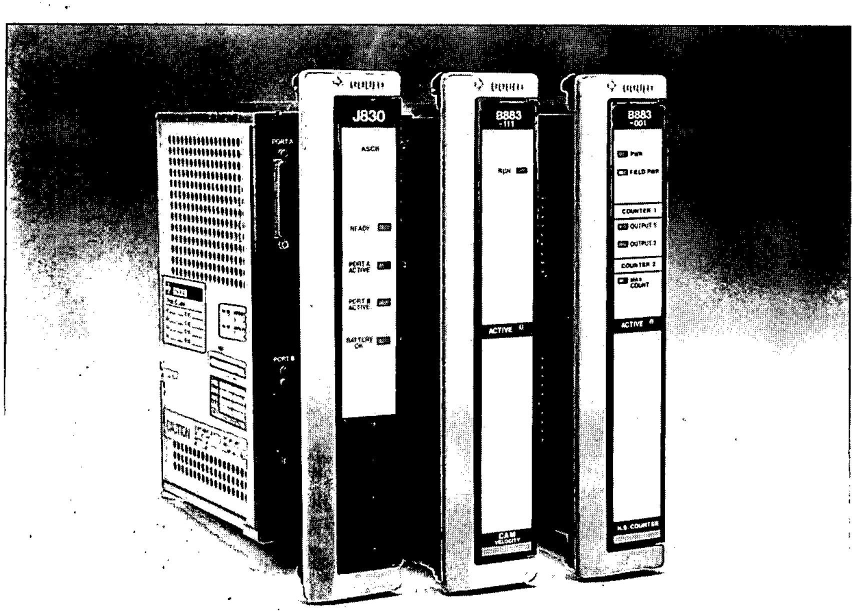
Superbly packaged, the 884A is an integrated, modular design, which provides maximum flexibility and ease of use. The basic system consists of the Controller, Power Supply, Input/Output Modules, and Programmer.



405 over 57

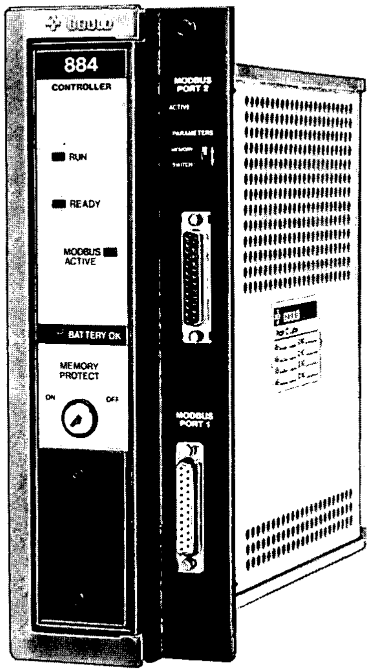
At the heart of the system, an 884A Controller module executes ladder logic and controls all other executive functions. An integrated, modular P884 Power Supply provides the necessary power to operate the 884A Controller and a large complement of I/O modules. The Power supply is jumper-selectable to either 120 VAC or 240 VAC while an optional power supply can be used to supply 24 VDC. Gould offers I/O modules in a wide range of voltages to support applications such as Discrete ON/OFF control and Analog Process control. For sophisticated applications beyond those normally associated with programmable controllers, Gould provides specialty I/O modules such as a High-Speed Counter, ASCII, CAM emulator, and Thermocouple.

The flexibility associated with 884A Systems extends to Program Loaders used to enter and document ladder logic programs as well as monitor and debug the system. Two Programming devices are currently available. The P190 Program Loader is designed for applications in a harsh industrial environment while the IBM Compatible Personal Computers can be used with the 884A programming software set for applications which require greater system flexibility. Both offer powerful, yet user-friendly programming, monitoring, and I/O fault diagnostic display. Gould currently supports the IBM XT, IBM PC, IBM AT, Compaq, Compaq Plus, and the Compaq 286.



406

II. 884 Controller Module



The 884A Controller Module controls all operations, including logic solving and system supervisory control functions.

The 884A processor can control a maximum of 2048 I/O bits. It allows users the flexibility to allocate Discrete and Register I/O among 1024 Input bits and 1024 Output bits. Each Discrete I/O point uses only 1 bit, while 16 bits are used per Register I/O Point.

For Discrete I/O, the 884A Processor can address 256 Discrete Inputs and 768 Discrete Outputs. Should the system require more than 256 Discrete inputs, it is possible to bring discrete signals into the input registers.

If Analog I/O is required, then the 1024 Input bits can be used as 64 Input Registers, while the 1024 Output bits can be used as 64 Output Registers. The 884A I/O structure also allows Analog I/O multiplexing. A maximum of 448 Analog I/O points can be multiplexed. Combinations of discrete and Analog I/O can also be addressed.

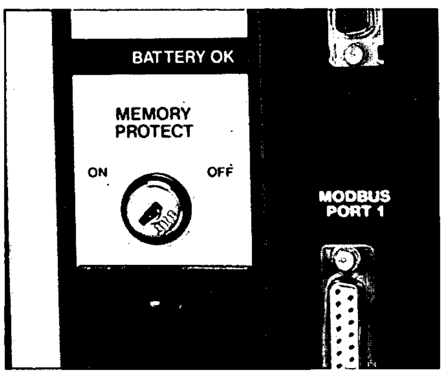
A battery-backed CMOS RAM user memory is available in 2.0k, 3.5k, or 8.0k, 16-bit words. There are no hidden reductions in user logic. Configuration of the I/O does not reduce user logic. One node programmed is typically one word used. In fact, an additional 1k words of memory for data storage comes standard with the 884A Processor.

Communications with mainframe computers or other devices can be handled using *Modbus*®, Gould's Local Area Network, integral to every 884 Processor. A second *Modbus* port provides an option for simultaneous communications with a *Modbus* Master and/or a programming device.

407 over→

Unique to the 884A Programmable Controller is the Built in Real-Time Clock/Calendar. It can provide year, month, day, hour, minute, and seconds data. A battery-backed capability continues timing during power loss. In addition, it interfaces with the J830 ASCII Module to generate reports tagged with time and date.

A four-level password system, in software, protects the 884A System from unauthorized access by allowing varying degrees of access to the system. In addition, a memory protect key-switch is a hardware mechanism which prevents unauthorized access to memory.



The CPU integrates three dedicated microprocessors. A 16-bit, Intel 8086 processor executes ladder logic with a typical scan time of 20 msec per 1k user logic for close control of the application. Another Intel 8051 microprocessor supervises communications between the Controller and the I/O modules to insure a steady flow of accurate data along the I/O bus. Because these two microprocessors operate in parallel, there is minimal time penalty for updating the I/O status, so that I/O throughput time is optimized. The third microprocessor, another Intel 8051, supervises all *Modbus* communications between the 884A Controller and any intelligent external devices, such as the program loader or a host computer. A fourth microprocessor is incorporated when the optional 2nd *Modbus* port is purchased.

The 884A Processor incorporates extensive fault detection techniques to insure data integrity and proper operations. The system status indicators located on the Power Supply, controller, and I/O modules report any major failures. For example, Watch Dog Timers monitor data flow between system components, and shuts down portions of the system that do not receive data wtihin a predetermined period of time. The Cyclic Redundancy Check (CRC-16) tests the accuracy of the data communications between the I/O modules and controller. Many other confidence tests provide rapid response to problems each logic scan.

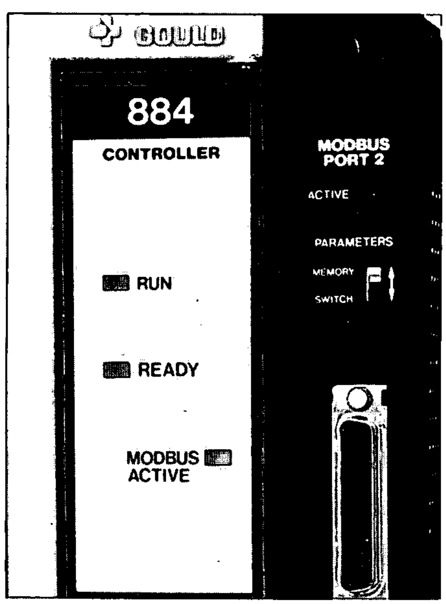
Four LED indicators on the controller's face plate provide a quick review of system status as follows:

READY — system components functioning properly.

RUN — controller solving ladder logic.

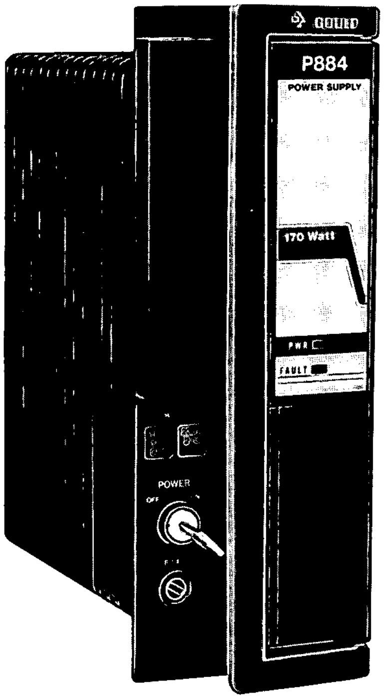
BATTERY OK — battery voltage within specification. If this indicator goes out the battery will retain memory for 14 days without external power, but must be replaced.

Modbus ACTIVE — *Modbus* port has received a valid *Modbus* communication. The *Modbus* active indicator shuts off at the end of every message transmission.



408

III. Power Supply



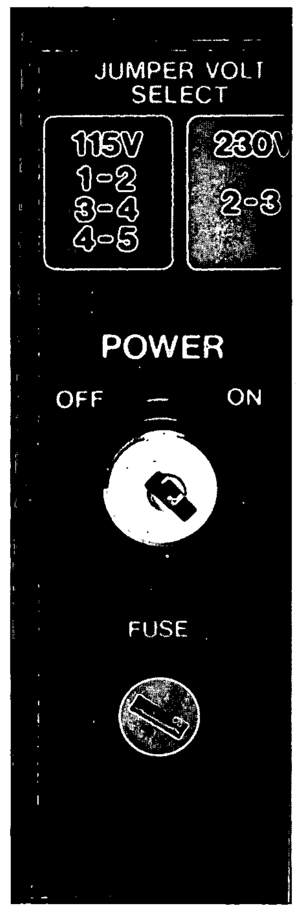
The Power Supply is located next to the 884A Processor at the left-most slot on the housing. Separately integrated, the P884 Power Supply provides sufficient power for the Controller and a full complement of I/O modules.

With a jumper switch, the user can instruct the Power Supply to accept either 115 VAC or 230 VAC at 50 to 60 Hz. The P802, 24 VDC Power Supply is also available. Designed to operate in a harsh industrial environment, the Power Supply can withstand temperature and humidity extremes, unstable and noisy power sources, sustained shock and vibration, and high frequency electromagnetic fields. Should the operating power fall below the specified range, the Power Supply will initiate a logical shutdown of the system. An output power limiter designed within the unit's circuits protects the Power Supply from a power overload. An indicator on the face of the Power Supply will indicate power overload. Rugged design eliminates the need for isolation transformers.

A key-switch located on the left hand side of the Power Supply allows power to be turned ON and OFF. LED's on the face of the Power Supply give a quick indication of the Power Supply's status. These indicators are:

PWR — Power Supply is functioning properly.

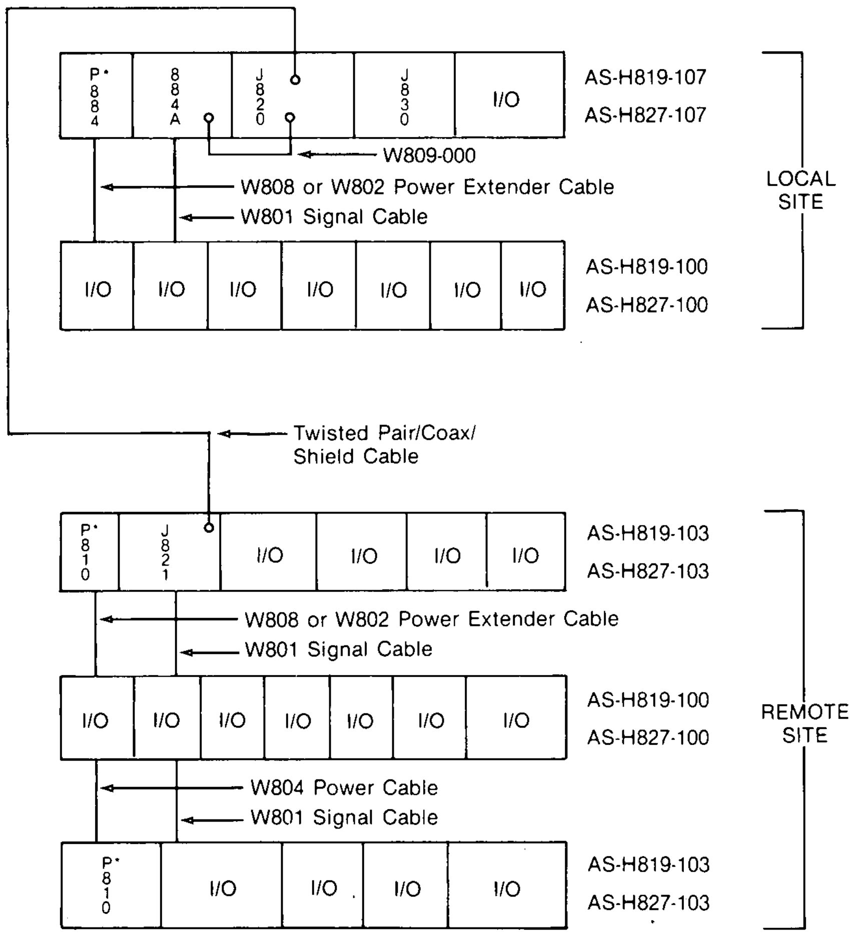
FAULT — Power Supply has exceeded specified limits.



409 over ⇒

TYPICAL SYSTEM CONFIGURATION

884 CONFIGURATION



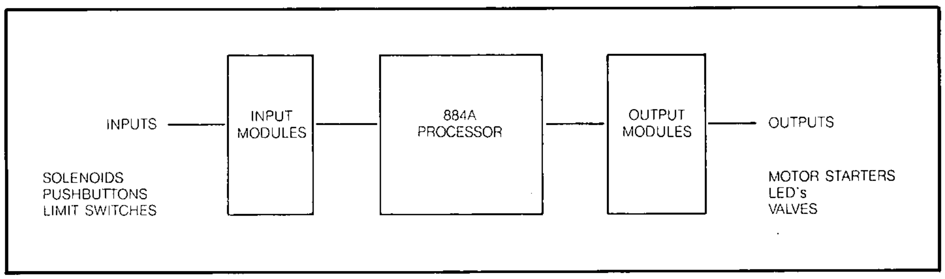
*Could be a P802 or P800

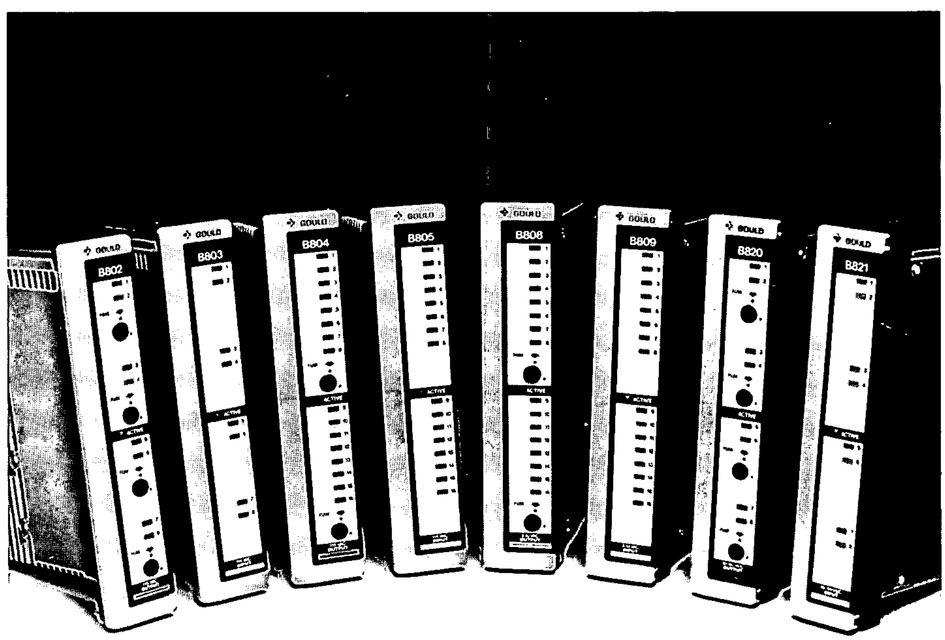
410

IV. I/O System

Gould's 884A Programmable Controller System interfaces with industrial equipment through the 800 Series Input and Output (I/O) Modules. Input Modules receive data from input devices such as limit switches, push-buttons, and analog devices, and sends this data to the Processor. Output Modules send instructions to the devices being controlled, such as motor starters, solenoids, and valves.

The 800 Series family of I/O offers a full range of Discrete and Analog modules in many different voltages, housed in tough, drip-proof packaging. The field wiring of most I/O modules is to rigidly mounted terminal blocks, permitting the removal of I/O modules without disturbing the wiring or connectors. This system eliminates the stress on wires because they are never moved. Built for use in a harsh industrial environments, the 800 Series I/O meets IEEE 472 Surge Withstand Standards.





411 over →

Compact design allows the maximum number of I/O to be placed in the minimum amount of space. Any of the 4, 8, 16, or 32 point I/O modules can be integrated together to suit system requirements since any module, Discrete or Analog can fit into any housing slot for I/O configuration flexibility. 800 Series I/O also incorporates smart I/O modules such as the ASCII module, CAM module, and High Speed Counter for more sophisticated applications.

AC Outputs rated at 2 amps, maximum per output point, and 12 amps maximum per module directly drive such devices as solenoids, and motor starters. These AC Output Modules use high-current surge triac devices that have been derated to enhance module and system reliability. A front panel fuse protects each output module and can be changed without removing the module from service. Each fuse has its own blown-fuse indicator for maintenance ease.

I/O Software addressing, handled through the "Traffic Cop" function of the processor, eliminates the need for switch settings on the I/O housings. The Traffic Cop indicates, via the programming device, incorrect placement of I/O modules within the housing slots. If a mismatch is indicated by the Traffic Cop, the I/O Processor will discontinue communications to that slot. The I/O Processor maintains a module Health Status bit for every module which is available to user logic for system diagnostics. An optional mechanical keying scheme prevents improper module placement in the I/O housing while a color code allows quick module recognition.

Traffic Cop Display

884 PC I/O CHANNEL TRAFFIC COP

CHANNEL #: 01 RACK #: 01

SLOT #	MODULE TYPE	REFERENCE NUMBERS	DATA TYPE	MODULE DESCRIPTION
101	P800			POWER SUPPLY
102	884A			MAINFRAME CPU
103	884A			MAINFRAME CPU
104	B805	10001-10016	DISC	115 VAC 16-in
105	B804	00001-00016	DISC	115 VAC 16-OUT
106	B864	00017-00144	DISC	REG 8 CH OUT
107	B862	40001-40004	BINARY	REG 4 CH OUT

	BINARY BCD	DEL SLOT NEXT SLOT	PREV RACK NEXT RACK	PREV CHNL NEXT CHNL	GET CHNL	WRITE CHNL	

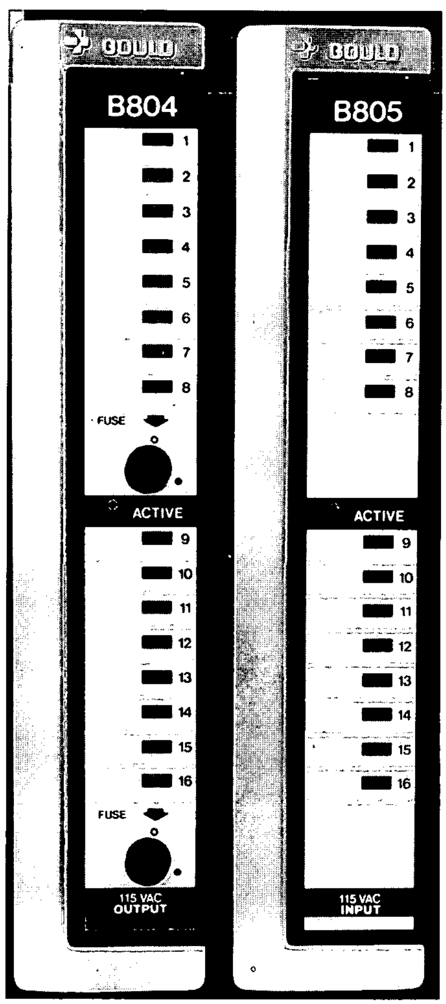
412

LED Indicators on each I/O module allow quick determination of the following module status.

STATUS — Reflects the ON/OFF state for each discrete I/O point. These indicators are connected to the field side of the module's circuitry and indicate the true status of the field device.

ACTIVE — Indicates the module is communicating properly with the controller.

BLOWN FUSE — Indicates a blown fuse. One indicator is provided for each fuse.



Individual data sheets are available for 800 series I/O modules and can be referred to for detailed information on keying, electrical specifications, and wiring.

LOCAL/REMOTE I/O

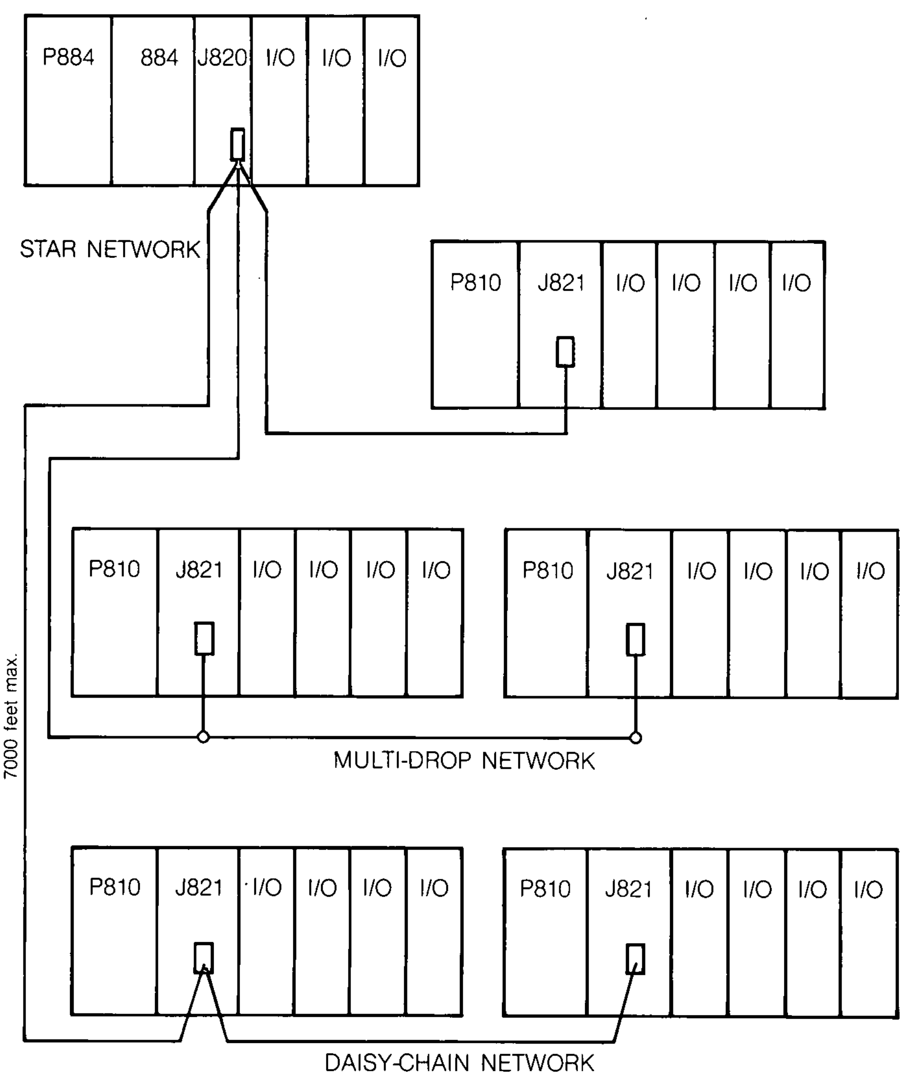
The 884A I/O system can accommodate both local and remote I/O configurations. Up to 5 I/O racks with a maximum of 32 I/O modules can be configured on an 884 system. Local I/O can be configured to a distance of 20 feet from the processor. The 884A Remote I/O system, which uses the J820 Remote I/O Driver and J821 Remote I/O Receiver, operates in conjunction with the 884A processor to allow up to 5 remote I/O drops, to be located up to 7,000 feet from the processor. Error detection and correction circuitry ensures network communications integrity.

The remote I/O system can be configured as a Multi-drop, Daisy-chained, Star-shaped network, or any combination of these. Many different types of cables can be used depending on individual requirements. A table of some of the available cables are listed.

413 over →

Cable selection:

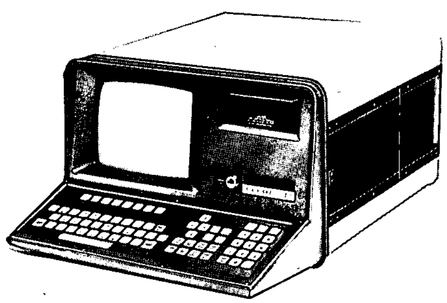
CABLE	DESCRIPTION	DISTANCE Feet	Meters
Belden 8760	Twisted Pair	2000	610
Belden 8795	Telephone Wire	2800	853
Belden 9152	Twisted Pair	3000	914
Belden 8227	Twin Axial	5100	1555
Belden 8263	RG59B Coaxial	6300	1920
Belden 9862	RG62A Coaxial	7000	2184



Remote I/O Drop Configuration
(Combination Network)

414

V. Programming



The 884A Processor uses standard Control Relay Ladder Logic. Either the P190, IBM Personal Computer, or compatible can program the 884A. The P190 industrial hardened program loader operates in harsh industrial environments where temperature extremes, and noisy power lines are common. The flexible IBM Personal Computer can be used with 884A programming software as a program loader. This also allows the program loader to handle color graphics, data acquisition, and many other functions.

P190 PROGRAM LOADER
The P190 program loader can program, edit, monitor, and list the 884A users program. A full 9″ CRT screen allows the display of a 10 column by 7 row matrix of contacts relays, and function blocks. Built-in programming flexibility allows multiple special function blocks to be programmed on the same screen.

Advanced editing capabilities allow easy programming, debugging, and trouble-shooting. The Element Editor allows contacts and elements to be entered directly into the contoller memory. The Network Editor allows an entire ladder network to be created off-line before it is loaded into the 884A processor. Once developed, the network editor can be used to insert new rungs into existing ladder logic or replace existing ladder logic.

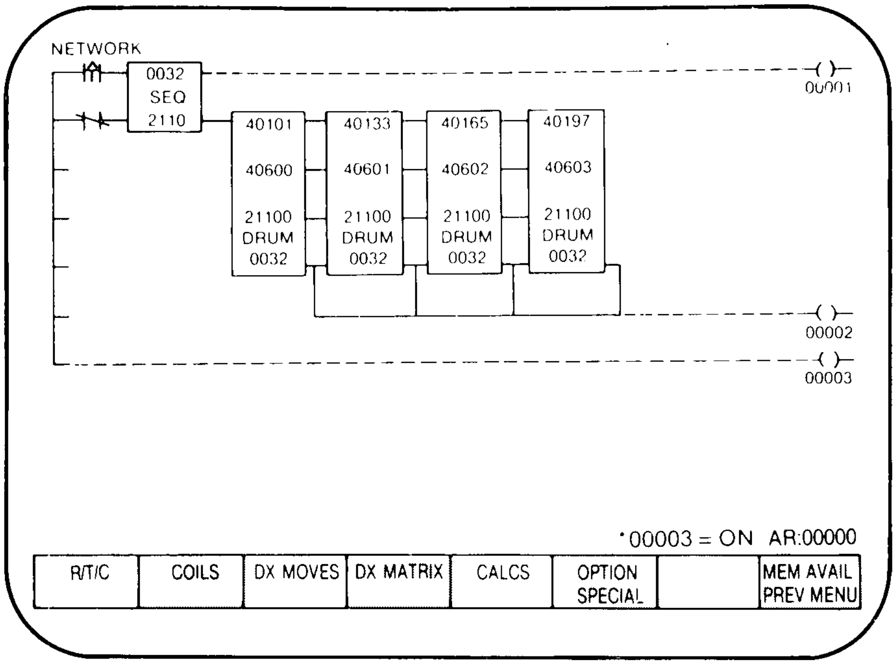
Gould's P190 also provides system monitoring capabilities. The status of individual contacts and coils are easily recognized with different light intensities to indicate power flow. Register data can be accessed for monitoring or editing as the system may require. Users can verify proper operation of I/O modules at a glance with diagnostic features such as the I/O Module Health Screen which identifies faulty I/O modules.

IBM PROGRAMMING
In addition to the P190, users can also program the 884A processor with an IBM Personal Computer, an IBM compatible, or the Gould IM 1062 Indusrial Graphics Terminal. These significantly enhance the flexibility of the program loader to handle tasks beyond those normally associated with program loaders. Software programming allows the IBM Compatible Personal Computer to perform all the functions of the P190 in addition to color graphics and data acquisition capabilities. The Annotated Ladder Lister allows the user to document the contacts and coils of the ladder logic program with mnemonics. This simplifies the task of programming and trouble-shooting by allowing the user to label a contact or coil with a real world device mnemonic. IBM Compatibles can also be used to program the 884's J830 ASCII module.

884 PROGRAM INSTRUCTIONS
The 884A instruction set incorporates 35 powerful individual special functions that allow the implementation of a wide variety of discrete or batch control applications. For instance, the Skip function can reduce scan time by allowing seldom-used sections of ladder to be skipped. Signed Double-precision math functions allows programming power for batch applications unmatched by other medium-sized processors. The Drum Controller allows sequential-type control over multiple points and can be cascaded to create the equivalent of 144 lobes by 99 steps of a mechanical drum. These special functions and others, such as ASCII Transmit/Receive, allow the user to go beyond the realm of standard relay replacement.

884A Special Functions
Relay Replacement Instructions
Relay Contacts (NO, NC)
Transitional Contacts (Positive; Negative)
Coils (Standard; Retentive)
Timers (1.0, 0.1, and 0.01 second time bases)
Counters (Up, Down)
Dual Input Latch

415 over →



The powerful new drum control function appears as a compact instruction for clear understanding of integrated logic. (Note: Drums can be cascaded creating the equivalent of 144 lobes by 99 steps of a mechanical drum.)...

P190/884	SOURCE	CONTACT · 0000	0000	0111	1111 ·	HEX
DRUM DISPLAY	REG NO	1234	5678	9012	3456	
	40101	STEP 01 · 1000	0000	0000	0000 ·	8000
DRUM IN	40102	02 · 0100	0000	0000	0000 ·	4000
NETWORK	40103	03 · 0010	0000	0000	0000 ·	2000
001	40104	04 · 0001	0000	0000	0000 ·	1000
	40105	05 · 0000	1000	0000	0000	0800
CONTACT	40106	06 · 0000	0100	0000	0000 ·	0400
DESTINATION	40107	07 · 0000	0010	0000	0000 ·	0200
40600	40108	08 · 0000	0001	0000	0000 ·	0100
	40109	09 · 0000	0000	1000	0000 ·	0080
POINTER	40110	10 · 0000	0000	0100	0000 ·	0040
21100	40111	11 · 0000	0000	0010	0000	0020
	40112	12 · 0000	0000	0001	0000	0010
SIZE	40113	13 · 0000	0000	0000	1000	0008
32	40114	14 · 0000	0000	0000	0100	0004
	40115	15 · 0000	0000	0000	0010	0002
	40116	16 · 0000	0000	0000	0001	0001

AR:00000

UPDATE	GET STEP	ENTER HEX	SET BIT	CLEAR BIT			RESTORE

...or as a detailed full-screen display for simple programming.

Single Precision Math
Addition
Subtraction
Multiplication
Division
Test (Binary compare >, =, <)

Signed Double Precision Math
Addition
Subtraction
Multiplication
Division

Stepping Control Instructions
Sequencer
Drum Controller (16 bit × 99 steps)

Data Transfer Instructions
Table-to-Register Move
Register-to-Table Move
Block Move

Bit Handling Instructions
Bit Shift (Left; Right)
Bit Sense
Bit Modify

I/O Monitoring Instructions
Reference Variable Health
I/O Module Health

Skip Instruction

Calendar Clock

416

VI. World Wide Support Services

In 1969, Gould introduced the industry's first Programmable Controller, the 084, which is still being serviced today. The 884A Programmable Control System includes that long-term commitment to professional support services. A working partnership is established to meet the needs of customers, not only at the time of purchase but also for years to come.

Field Service Product Engineers work closely with all product development groups to ensure that serviceability and reliability are engineered into every product. On call hardware service and installation start-up support are available locally and world wide. The Field Support Center provides around-the-clock, toll-free answers to hardware and software questions and problems.

To reach our Field Support Center call:

1-800-GOULD IA

Gould's support organization can supply application and maintenance training, interactive video, and complete user and maintenance documentation. Gould provides a broad range of courses. These courses are offered at most local sales offices, the Gould Customer Training Center in Salem, New Hampshire, or at a customer facilities.

Your local Sales or Service Representative can help you get full details regarding your partnership with Gould Industry Automation Systems Division.

417 over →

884 — PRODUCT FAMILY

CONTROLLER

Mainframe

CPU	MEM	*Modbus*
884A-101	2.0K	1 PORT
884A-201	3.5K	1 PORT
884A-301	8.0K	1 PORT
884A-111	2.0K	2 PORT
884A-211	3.5K	2 PORT
884A-311	8.0K	2 PORT

Power Supplies

P884-001	PRIMARY	115/230 VAC
P802-001	PRIMARY	24 VDC
P810-000	AUX/ REMOTE	115/230 VAC

Module Housings

H819-107	PRIMARY - (Options)	4 SLOT
H819-103	PRIMARY	4 SLOT
H819-100	SECONDARY	7 SLOT
H827-107	PRIMARY - (Options)	8 SLOT
H827-103	PRIMARY	8 SLOT
H827-100	SECONDARY	11 SLOT

Housing Cables

POWER		
W808-002	W804-002	1.5 FT
W808-006	W804-006	5.5 FT
W802-012	W804-012	12.0 FT

SIGNAL	
W801-002	1.5 FT
W801-006	6.0 FT
W801-012	12.0 FT

Remote I/O

J820-001	DRIVER
J821-001	RECEIVER
W809-000	CABLE

TECHNICAL MANUALS

884PC Technical Library **PI-884A-LIB**

Includes
- System Planning and Installation
- Programming Guide / P190
- Tape Loader User's Guide
- Maintenance Manual
- Programming Reference Card

Remote I/O System Installation and 2nd Operation **PI-J820-001**

NOTE: User manuals included with all other hardware and software.

INPUT/OUTPUT MODULES

Discrete

INPUT		
B803-008	8 pt	115 VAC
B805-016	16 pt	115 VAC
B809-016	16 pt	230 VAC
B817-116	16 pt	115 VAC ISOL
B817-216	16 pt	230 VAC ISOL
B821-008	8 pt	10-60 VDC
B825-016	16 pt	24 VDC TH
B827-032	32 pt	24 VDC TH
B829-116	16 pt	TTL/CMOS
B833-016	16 pt	24 VDC TL
B837-016	16 pt	24 VAC/DC
B849-016	16 pt	48 VAC/DC
B853-016	16 pt	115 VAC/ 125 VDC
B863-001	4 ch	TTL/CMOS
B865-001	8 ch	TTL/CMOS
B869-001	8 ch	TTL/CMOS CH-SELECT
B881-001	16 pt	24 VDC Latched

OUTPUT		
B802-008	8 pt	115 VAC
B804-016	16 pt	115 VAC
B808-016	16 pt	230 VAC
B810-008	8 pt	115 VAC ISOL
B814-108	8 pt	Relay NO/NC
B820-008	8 pt	10-60 VDC TH
B824-016	16 pt	24 VDC
B826-032	32 pt	24 VDC TH
B828-016	16 pt	TTL/CMOS
B832-016	16 pt	24 VDC TL
B836-016	16 pt	12-250 V
B840-108	8 pt	Reed Relay NO/NC
B862-001	4 ch	TTL/CMOS
B864-001	8 ch	TTL/CMOS
B868-001	8 ch	TTL/CMOS CH-SEL

Analog

INPUT		A/D
B846-001	16 ch	Multiplexer-voltage
B846-002	16 ch	Multiplexer-current
B873-001	4 ch	4-20 ma/1-5V
B873-011	4 ch	−10V to +10V
B875-001	8 ch	4-20 ma/1-5V
B875-011	8 ch	−10V to +10V
B875-101	8 ch	Fast Analog
B883-200	10 ch	Thermocouple 10 CA
B883-201	8 ch	RTD

OUTPUT		D/A
B872-002	4 ch	4-20 ma/1-5V
B872-011	4 ch	±10V, ±5V

INTELLIGENT	
B883-001	High-Speed Counter
B883-101	ABS Encoder/CAM w/o velocity
B883-111	ABS Encoder/CAM w/ velocity
B884-002	Dual PID Loop Controller
B885-001	ASCII/BASIC

PERIPHERALS

P190

- SW-PR8T-0TA PROGRAMMING SUPPORT MODULE
- SW-D08T-0TA DOCUMENTATION SUPPORT MODULE
- SW-CS8T-0TA CONTROLLER SUPPORT MODULE

IBM PC/XT

- SW-PR8D-1DA PROGRAMMING SUPPORT MODULE
- SW-DO8D-1DA DOCUMENTATION SUPPORT MODULE
- SW-CS8D-1DA CONTROLLER SUPPORT MODULE

DEC PDP11 Software

- SW-DO8R-1XA DOCUMENTATION SUPPORT MODULE
- SW-APPR-1XA COMMUNICATIONS HANDLER

Communication Hardware

- J878-000 *Modbus* MODEM

ASCII Module

- J830-001 884 ASCII MASTER

LITERATURE

Data Sheet

- MC-884A-001

Product Description

- MC-0884-040

418

VII. Specifications

Controller

I/O Capacity	1024 Input Bits Max./1024 Output Bits Max. 768 Discrete Coils Max. 256 Discrete Inputs Max. 64 Input Registers Max. 64 Output Registers Max.
Memory Size	2k, 3.5k, 8k Battery-Backed CMOS RAM
Word Size	16-Bit words
Data Storage	864 Holding Registers Max.
Remote I/O	5 Remote drops max.; 7000 ft. max. in Star, Daisy-Chain or Multi-drop format; Twisted-pair or Co-axial cable.
Scan Time	20 msec per 1k memory (typical)
Diagnostics	Watchdog Timers, CRC-16, and Continuous Confidence Testing, Module Health Status
Communication Ports	2 Modbus interface ports — RTU or ASCII
Indicators	RUN, READY, Battery OK, Modbus Active indicators
Keyswitches	ON/OFF and Memory Protect
Power Supply Indicators	OVERLOAD, POWER indicators

Instruction Set

Basic Instructions	Relays, Timers, Counters, and Transitional Contacts
Single/Double Precision Signed Math	Add, Subtract, Multiply and Divide
Data Transfer	Block Move, Table-to-Register Move, Register-to-Table Move, Drum Controller
Bit Operations	Sequencer, Bit Shift Left/Right, Bit Sense, Bit Modify
Special Functions	Skip, Single-Double/Double-Single Precision Conversion, Real Time Clock, Latches, I/O OK and I/O Status, TEST (compare), Optional ASCII Read/Write

Programmer P190 Program Loader, IBM Compatible Personal Computer and Controller Support Module Software

Power Requirements

AC Power	115/230 VAC; 167 Watts
DC Power	24 VDC; 167 Watts
Frequency	47-63 Hz
Line Spike	2500 V for 6 usec.

	115 VAC	230 VAC	24 VDC
Normal Voltage	95-135 V	190-276 V	21-29 V
10 sec. max.	80-150 V	164-300 V	
1 cycle max.	0-200 V	0-400 V	

419 over→

Environment

Temperature	
Operational	0 to 60°C
Storage	−40 to 80°C
Humidity	0 to 95% (non-condensing)
Shock	±15 g halfsine to 10 msec
Vibration	5 to 50 Hz, 0.625 g (0.005 inches max. peek-to-peek)
EMI	Meets IEEE 472-1974 Meets MIL-STD 461-B

Physical

Size	The 884 with power supply uses a total of 3 slots in the following 800 Series Housings: H819-103/107 H827-103/107
Weight	4.5 lbs. (2.04 kg)
Battery Life	35 weeks

Power Consumption
Power for the 884 processor is supplied by either the P884 or the P802 power supplies. The power required by the 884 is as follows:

Mainframe	+5 VDC	+12 VDC	−12 VDC
884 (1 *Modbus* Port)	4400 mA	50 mA	50 mA
884 (2 *Modbus* Ports)	5300 mA	100 mA	100 mA

420

TABLE OF CONTENTS

INTRODUCTION

The 484 Controller Applications Manual is the publication of the Modicon Applications Engineering Department. This manual is a collection of many of the program examples which were developed in response to specific problems or in the demonstration of the capabilities of the 484 Controller. This manual is made up from a series of general purpose application notes aimed at solving a specific problem. It is the intent of this manual to serve as a guide to the programmer in offering techniques which can be adapted to their specific program.

Before attempting to utilize any of the application notes shown in this manual, it is advisable for the reader to be familiar with the operation and basic programming techniques of the 484 Controller.

This manual is the first in a series and incorporates several of the previously published single page application notes. It is our intent to periodically update this manual with additional application notes.

422 over

Seal and Latch Circuits

If a momentary pushbutton is used to start a motor, pump, etc., it must be "sealed in" so that the pushbutton does not have to be depressed continually to keep the motor running. Conversely, if the motor is to be stopped, a momentary pushbutton would "unseal" the circuit. The ladder diagram for a simple start/stop circuit with a "seal" is shown in Figure 1.

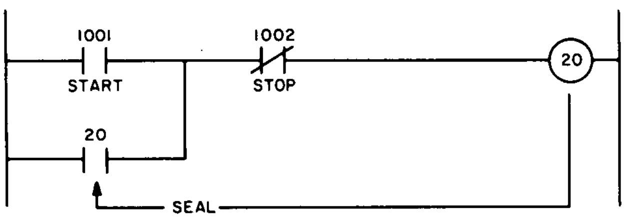


Figure 1. Seal Circuit

A LATCH circuit is the same as a SEAL circuit except that a latch coil is used (see Figure 2). The LATCH will remember the status of the coil through a power failure. The latch reference contact 20 will close on resumption of power and energize coil 20 (assuming the stop button is not depressed). This means that the operator does not have to re-start motors having a latch circuit after a power failure.

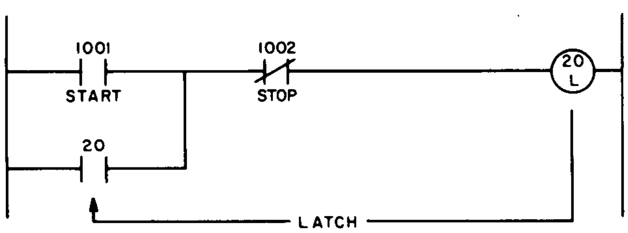


Figure 2. Latch Circuit

One Shot

A ONE SHOT is a signal that is ON for one scan of the controller whenever it is initiated. Transitional contacts (⊣↑⊢ or ⊣↓⊢) are one-shots that are available with the ENHANCED instruction set of the 484. To create a ONE SHOT without using transitional contacts, use either of the two methods shown in Figure 3 or 4.

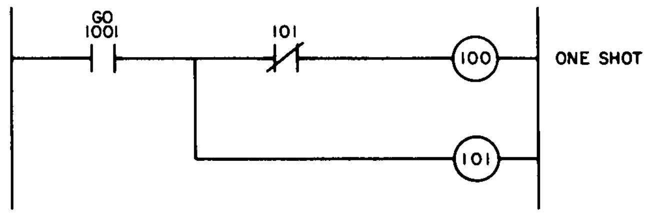


Figure 3. One-Shot from Coil 100

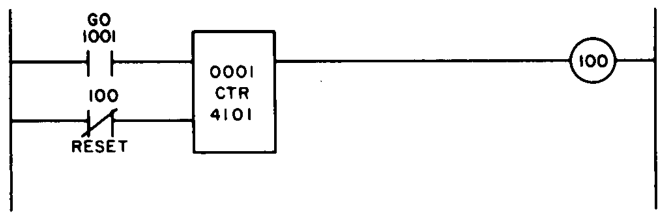


Figure 4. One-Shot Using Counter

Oscillator and Cycle Generator

An OSCILLATOR is a signal which is ON and OFF for a set amount of time. The simplest oscillator is one which is ON for one scan and OFF for one scan. It is programmed as shown in Figure 5.

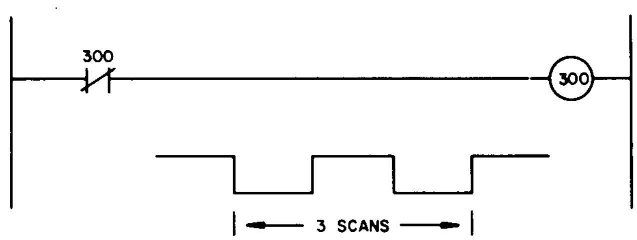


Figure 5. Oscillator

A variation of the one scan oscillator has a signal that will be ON for any designated period of time and OFF for some other designated time. It is called a cycle generator. Figure 6 details the circuit.

423

The cycle generator in Figure 6 is used to turn coil 301 ON for 30 seconds and OFF for 20 seconds.

Uses of the OSCILLATOR:

Scan Time Evaluator
Initiate Operations Every Other Scan
Fast Flashers

Uses of the CYCLE GENERATOR:

Annunciator Flasher
Energy Management (Duty Cycle Control)
Time Proportioning Process Control

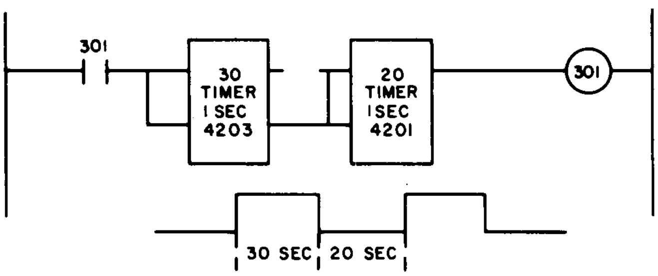


Figure 6. Cycle Generator

Dual Action Pushbutton

A DUAL ACTION PUSHBUTTON is one that performs two alternating functions. The first time it is depressed it will turn something ON. The second time it will turn it OFF. The third ON, etc. This can be accomplished with the ladder diagram in Figure 7.

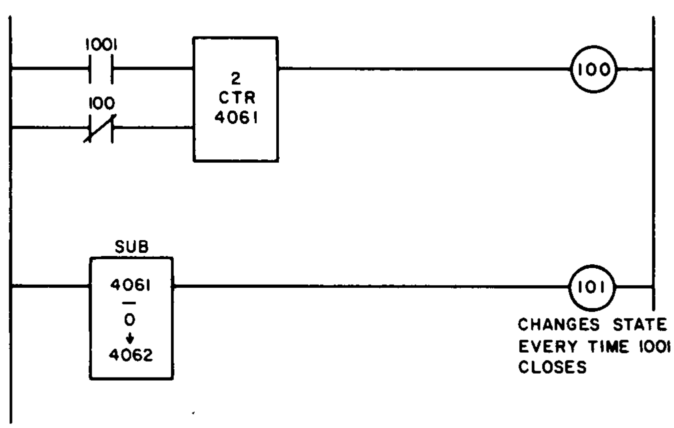


Figure 7. Dual Action Pushbutton

Retentive Shift Register

In a sample conveyor system, boxes enter the conveyor from the left, go through 5 stations and exit to the right. The stations have the necessary equipment to detect if a box is there (light cell, limit switch, etc.) and then perform functions such as loading the box with product, sealing and labeling the box, etc. If no box is detected, we do not want to load the empty space with product, seal it, label it, etc.

Example:

The ladder diagram program shown in Figure 8 will operate a 5 station shift register. If more stages are desired, the same logic format is simply added to more networks. Notice that the stations are programmed in descending order starting with the last station in network 1 and ending with the first station in the last network. This is due to the timing and logic solving of the controller and is required to make the shift register work.

Two input signals are used. One to enter good or bad box data (Reference 1002 is energized for good parts) and one to shift (Reference 1001) good/bad information to the right to coincide with the actual conveyor position.

The CLOCK SIGNAL is nothing more than a one-shot (signal only ON for one scan) (see page 2) created every time 1001 (SHIFT) is depressed.

COILS 111-115 are used here as output coils which can be used to operate (or not operate) the equipment associated with each station. The LATCH is used when it is required to make the shift register RETENTIVE. If it is not desirable to maintain the status of the stations through a power failure (such as when the conveyor is manually advanced and the controller is off), regular coils are used.

Figures 9 and 10 show variations of this basic logic to allow for the control of more complex operations.

424

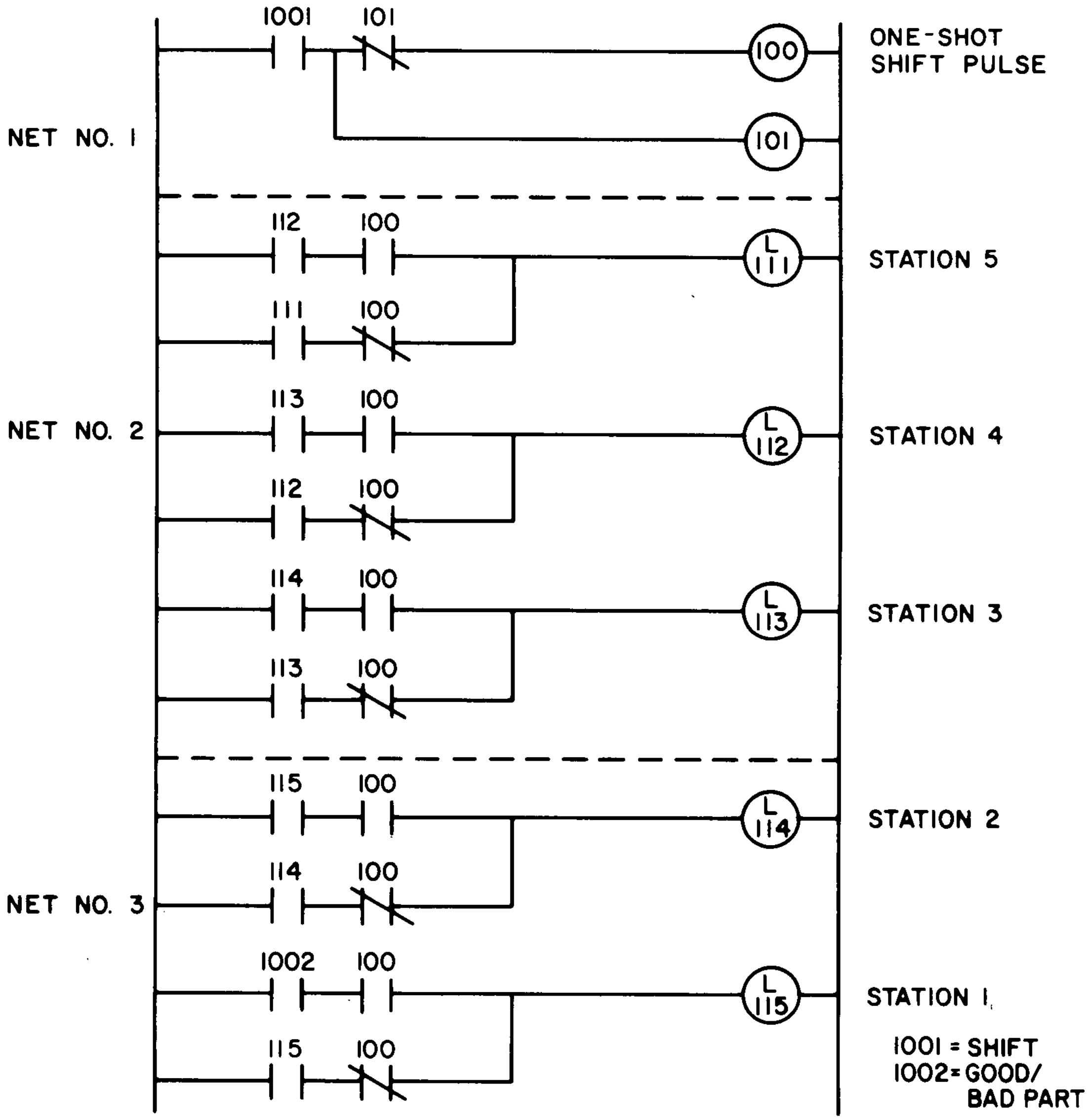


Figure 8. Retentive Shift Register

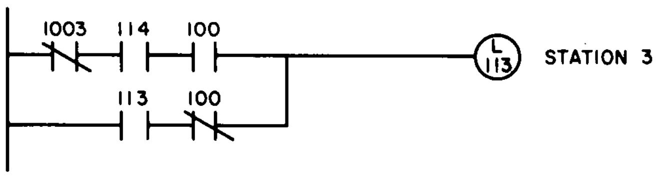


Figure 9. 1003 De-energizes when Good Box Becomes Bad at Station No. 3

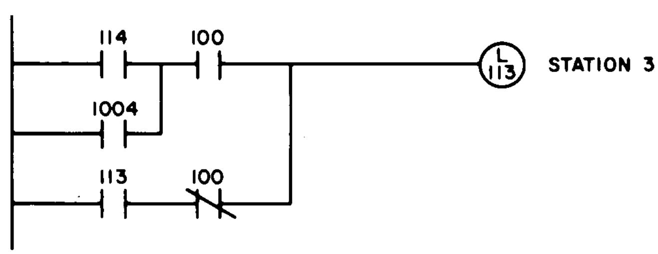


Figure 10. 1004 Energizes when Bad Box Becomes Good at Station No. 3

425

Alarm Point Monitoring

The programming below controls the operation of an alarm light. See Figure 11. When the alarm point (1009) transitions from open to closed (the alarm condition), the alarm light (303) begins to flash. When the alarm is acknowledged, the lower output (1003) is energized and the light comes on steady. When the alarm point is reset, the light goes out. If the alarm point comes on and then turns off almost immediately, the light will flash until acknowledged, at which time it will go out.

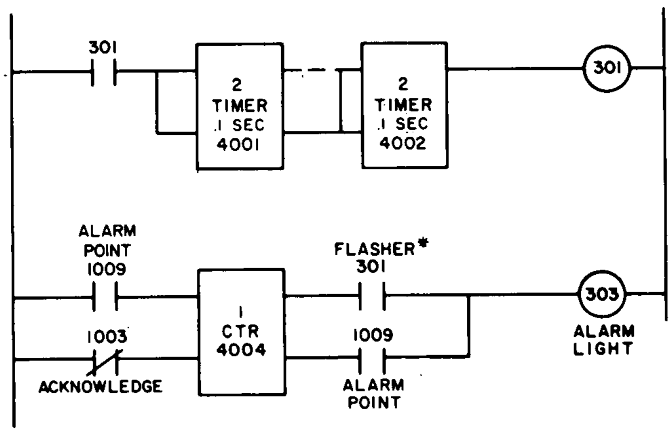


Figure 11. Alarm Point Monitoring

Scan Timer

A SCAN TIMER is used to evaluate the scan time of the 484 controller. See Figure 12.

An OSCILLATOR is used to generate a pulse signal, one scan ON and one scan OFF. The counter counts the pulses until the count reaches 500 transitions (1000 scans). At the same time, the timer times how long it takes the counter to count the 1000 scans. The timer is stopped when the counter reaches 500 transitions. Register 4041 will contain the scan time of the controller in milliseconds. The circuit can be placed in any unused network and later deleted when the scan time is verified.

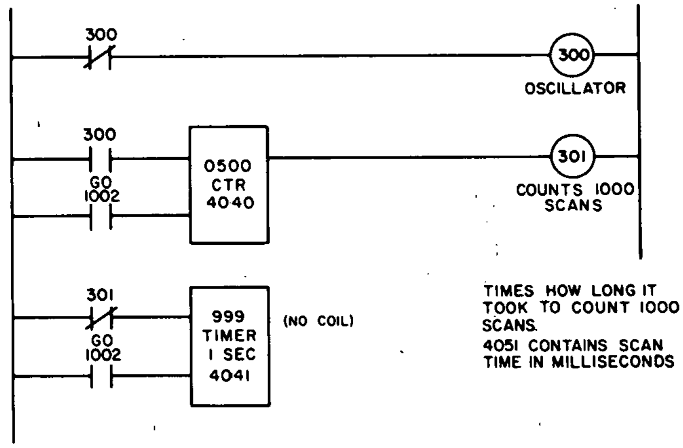


Figure 12. Scan Timer

Down Counter

A DOWN COUNTER will start at a designated preset and count DOWN on each succeeding signal. When the counter decrements to ZERO, a coil will go ON. The ladder diagram is found in Figure 13.

The DOWN COUNTER will use a regular UP COUNTER to count the GO signal transitions. The RESET will merely clear the current count at any time. The SUBTRACT function will subtract the preset (0073 in this case) from the current count (in 4061) and place the DOWN COUNT result in 4062 as an absolute value. Coil 60 will come ON when the preset has been reached and when the down counter has counted down to ZERO.

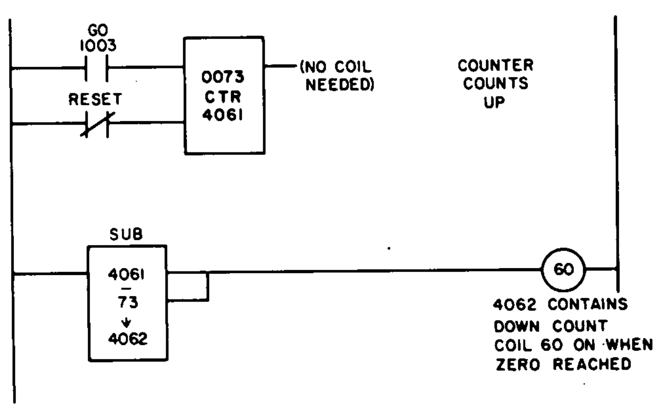


Figure 13. Down Counter

426 over →

Set Point Control

Set Point Control in its simplest form compares an input value, such as an analog or thumbwheel inputs, to a set point value. A discrete output signal is provided if the input value is less than, equal to or greater than the set point value. The subtract function with its 3 outputs can be used for set point control.

The first example, shown in Figure 14, can be used to detect when the input data is greater than or equal to the set point. The input data will be the contents of register 3001 and it will be compared every scan with the set point, which will be the contents of register 4050. The result of the subtraction will be the contents of register 4062, but this result will not be used in the set point.

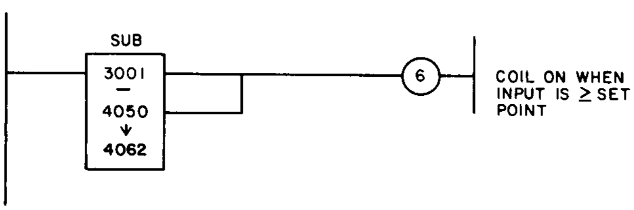


Figure 14. Output when Input Value ≥ Set Point

The second set point example, shown in Figure 15, can be used to detect when the input value is less than or equal to the set point.

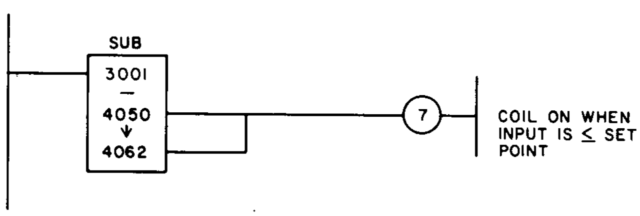


Figure 15. Output when Input Value ≤ Set Point

The final set point example, shown in Figure 16, is monitoring an input within a deadband, high and low set points. This logic will provide signals when the input is greater than high set point and less than low set point.

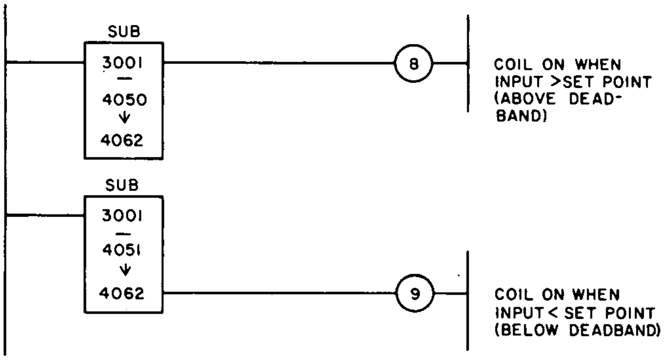


Figure 16. Set Point Control with Deadband

Double Precision Arithmetic (Addition, Subtraction, Multiplication)

In many applications there is a need to perform arithmetic functions on numerical values which are greater than 999 (the maximum content of a holding register). The double precision arithmetic techniques shown below allow the user to manipulate six-digit numbers.

Double Precision Addition

This routine takes the sum of two six-digit numbers in registers 4001-4002 and 4003-4004, both double precision, and stores the seven-digit result in registers 4005-4007, triple precision. This routine is shown in Figure 17.

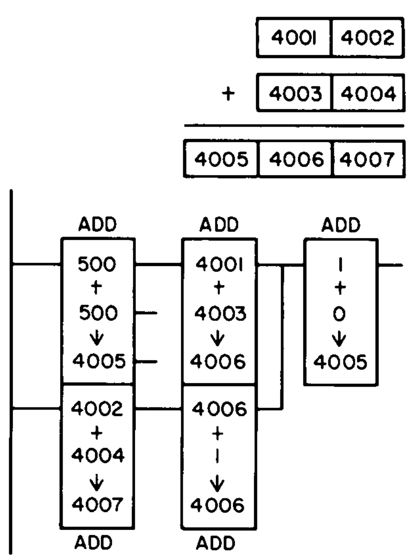


Figure 17. Double Precision Addition

4127

Double Precision Subtraction

This routine takes the difference between two six-digit numbers in registers 4001-4002 and 4003-4004, both double precision, and stores the six-digit result in registers 4005-4006, in this example shown in Figure 18.

If the difference is negative, the result assumes a borrow from the millions digit of the minuend.

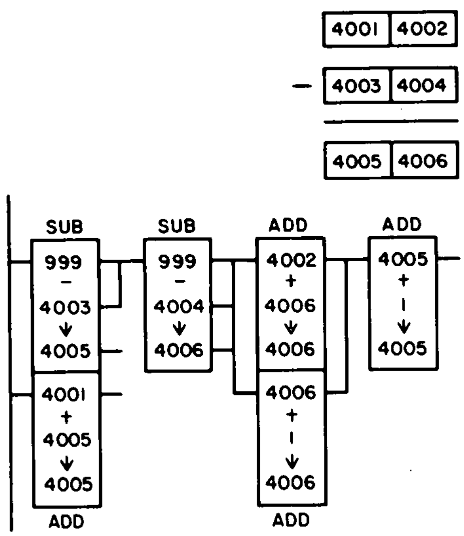


Figure 18. Double Precision Subtraction

Double Precision Multiply

This routine takes the product of two six-digit numbers. The two factors are placed (double-precision) in registers 4001-4002 and 4003-4004. The product (up to 12 digits) is returned in registers 4051-4054, for this example, as shown in Figure 19.

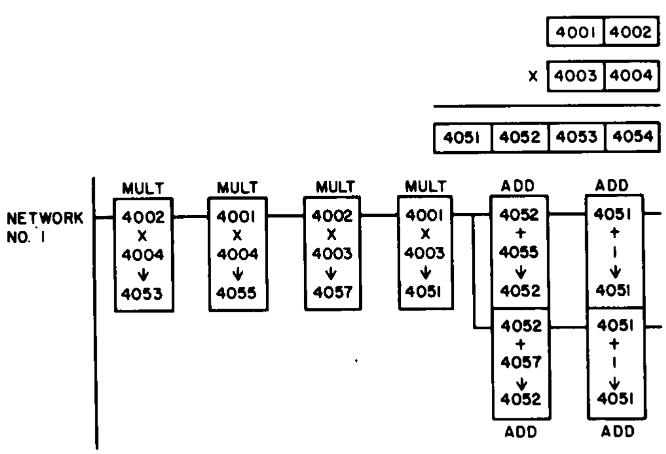


Figure 19. Double Precision Multiplication (Cont)

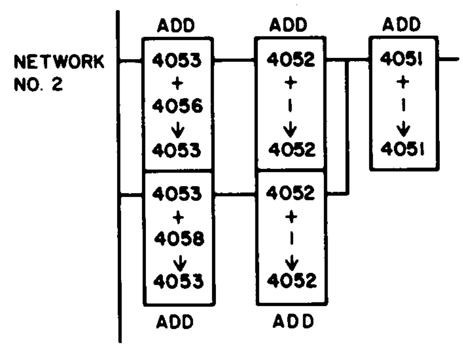


Figure 19. Double Precision Multiplication (Cont)

Extracting Square Root of a Number

The Square Root program takes a double precision number, in this example the contents of registers 4001-4002, and produces a single precision value, register 4050, as its square root. This function is useful for applications such as measuring the differential pressure across a flow element and extracting the square root of the pressure to determine flow.

The method shown in Figure 20, to extract the square root, is consecutive iterations. The logic will continually make guesses at the square root based on the formula 1/2 (n/x + x) where n is the previous best guess.

The first three functions, two subtractions and one addition, check that the number from which the logic does the square root extraction is based on the above formula.

This function is continuous with no control contact and once a value is placed into registers 4001-4002, the final square root function will appear in register 4050 in no more than 10 scans of the controller.

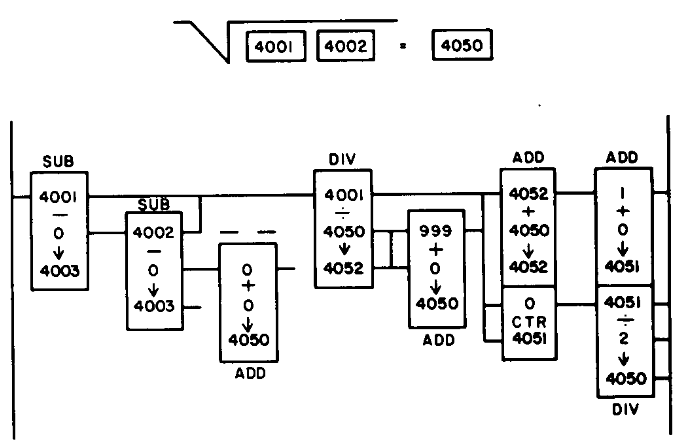


Figure 20. Square Root Extraction

428 over→

BCD Coded Thumbwheels to Replace 9's COMPLEMENT Thumbwheels

When entering BCD data into a 484 controller via a B571 (input BCD Multiplexer), it is recommended that 9's COMPLEMENT coded thumbwheels be used for decoding. It is possible, however, to use straight BCD coded thumbwheels. This following program is used to perform this conversion. For example, a BCD coded switch, wired to the B571, is set at 751. The BCD coded switch supplies the following code (MSB to LSB) to the module: 0111 0101 0001. This value is interpreted, and loaded into an input register, by the B571 module, as 248.

Since 999 - 248 = 751, incorporating the following logic into the program would allow the use of BCD coded switches with the B571:

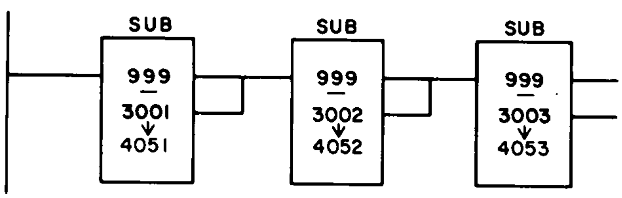


Figure 21. BCD Thumbwheel Conversion

One subtract plus one holding register, would be required for each three digit thumbwheel; and all other logic requiring use of the thumbwheel information would reference the appropriate holding register, and *not* the input register.

Operator Setting and Monitoring of Register Contents

In many cases, it may be desirable to give the operator the ability to enter numerical information into registers in a 484 Controller with the Enhanced II instruction set. This data entry can be through two thumbwheels, one to enter the three digit data and the second to enter the stop or location where the data is to be entered. In addition, the operator can have a three digit LED display for monitoring the contents of registers and a keyswitch to control when data is loaded. The following sample logic, shown in Figure 22, will allow the operator to load a table of 50 registers, starting with register 4101 and ending with 4150. The first step would be for the operator to select which register in the table will receive the new data, a range of 1 through 50. This would be accomplished by connecting the first thumbwheel to inputs 1001-1012, the result of which is stored in register 4050. Register 4050 is the pointer for the register to table move and 100 is added to its contents in order for loading of register to start at 4101.

The second step is to select the new data to be entered. The associated thumbwheel switch is connected to inputs 1013-1024 and its result is stored in register 4059 which is the source for the register to table move.

The third step is to energize the enter signal, input 1001, which should be a key switch to some security device in the system.

The last step allows the operator to monitor the contents of a register in the table as selected by the step thumbwheel. A three digit LED display can be connected to outputs 1-12, and the convert function will drive the display.

As an alternative, the thumbwheels can be connected to input registers 3001 and 3002 via the B571 input multiplexer module, and the LED display can be connected to output register 4001 via the B570 output multiplexer module. This would eliminate the requirement for the three convert functions.

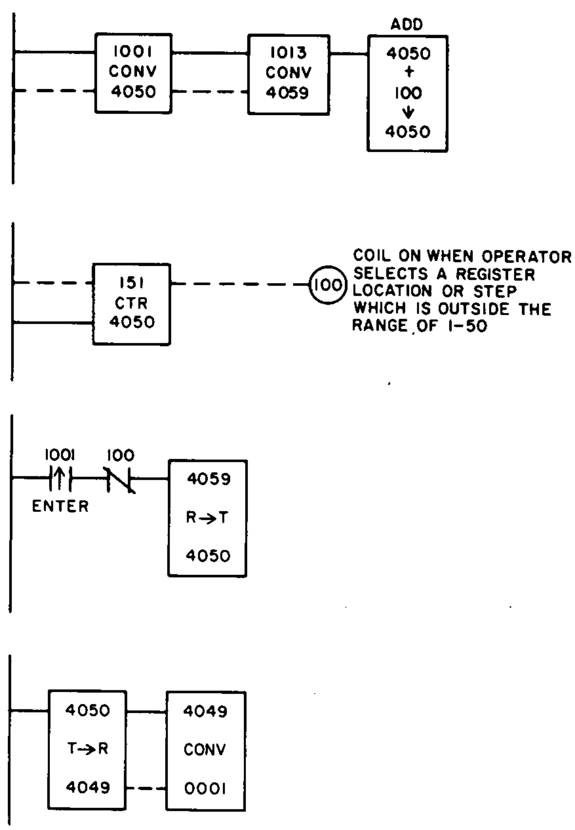


Figure 22. Setting and Monitoring Registers

429

Multiple Timers Using Two Registers

A cyclical process often requires a full cycle to be completed on a time basis. Within this cycle time, many operations may be required to be performed, also on a time basis. The total cycle and individual operations may be performed by using one timer per operation. This, however, would use up at least one register per timer plus the elements to initiate and reset each timer. An alternate method may be used to keep track of the time of the individual events by using a master timer that times the total cycle. As the timer times increments, a subtraction function determines when an individual operation is to be performed based on the master timer. See Figure 23 for a typical timing diagram and Figure 24 for the associated logic.

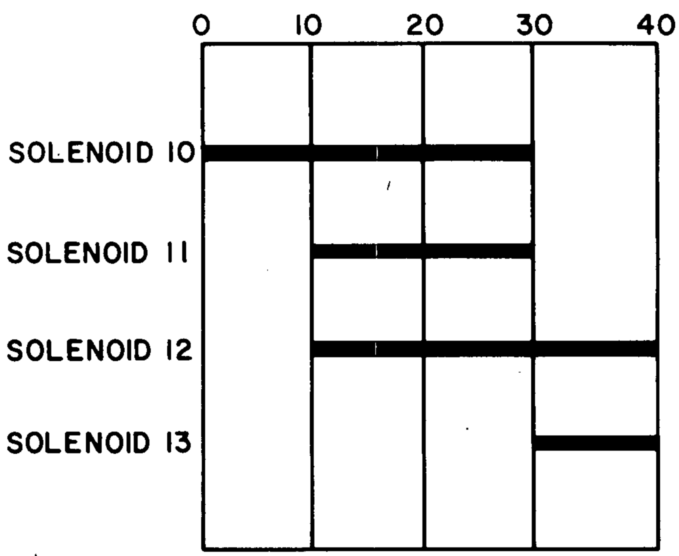


Figure 23. Cycle Timing Diagram

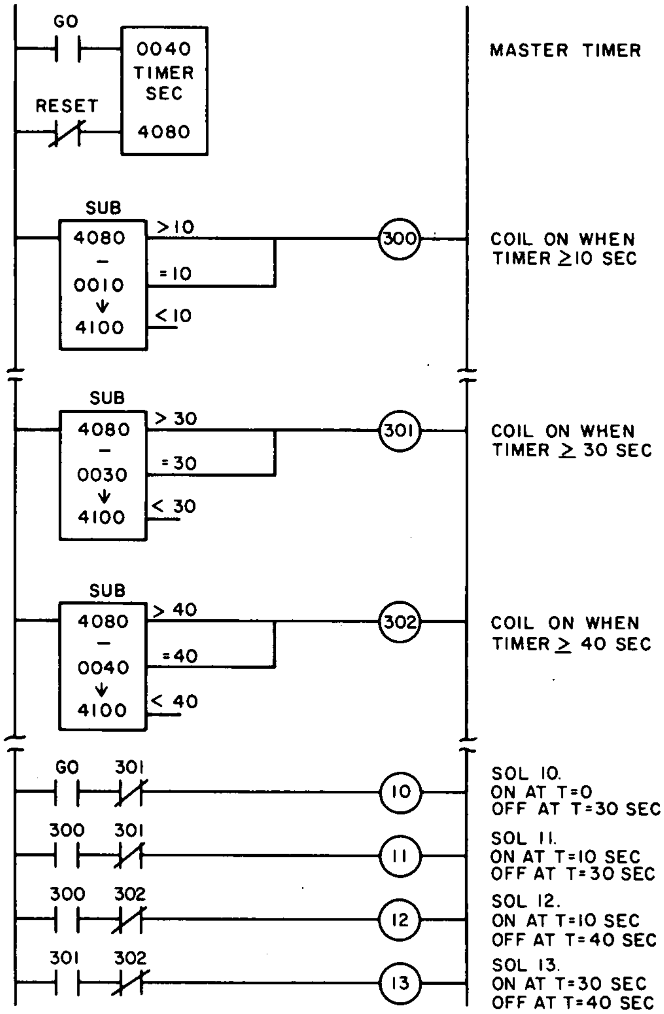


Figure 24. Cycle Timing Logic

430 → over

Referencing Last Available Registers (Input and Output)

In programming for the 484, certain calculations, specifically multiply and divide, require two consecutive registers. A statement as shown in Figure 25 would not be accepted, since the digit would require a register beyond 3032, which does not exist.

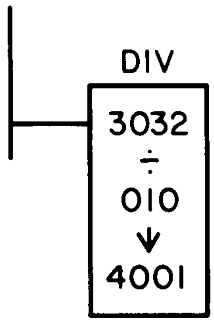


Figure 25. Unacceptable Division

No calculate can be programmed to accept the last available register (input or output) in a 484 PC. While Figure 26 will be acceptable, Figure 27 will not be.

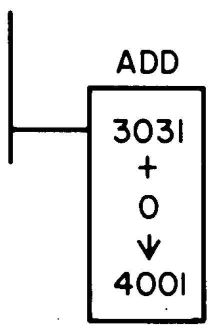


Figure 26. Acceptable Addition

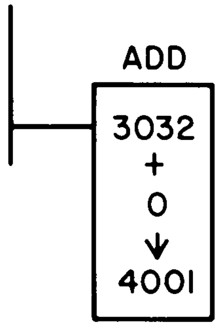


Figure 27. Unacceptable Addition

Figure 28 shows programming to allow the contents of 3032 to be recovered. 4XXX is any available holding register. Network 1 loads a value 999 into the holding register. Network 2 acts to force 4XXX back to equal 3032. The "preset" is 3032, and, since the holding register for a timer or counter cannot exceed its preset, 4XXX when loaded with 999, is forced back to whatever 3032 is. Now, whenever 3032 is required for a calculate, use 4XXX instead. The two networks (Figure 28) should be programmed together – no attempt should be made to use 4XXX between the two networks.

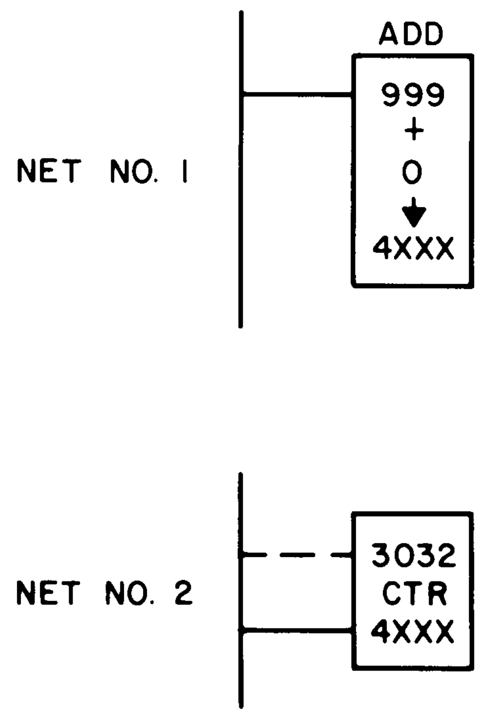


Figure 28. Recovering Contents of Register 3032

Note: The other 31 input registers (3001-3031) do not require this technique. While 3032 can't be used in any calculate, it can be used as timer or counter preset.

The last available holding register (except 4254 for a 484-165 (4K)) behaves in a similar manner; this technique may also be used for those registers.

431

Limiting Register Contents

The contents of a register may be limited to a specified range with the logic shown in Figure 29. For example, if a value in register 4001 is required to be held between the value 023 and 900, the logic would take separate actions to truncate the high limit to 900 and the low limit to 023. The high limit is truncated by placing 4001 as the "current value" register in a timer or counter. The low limit requires testing with a subtract line. If the contents in 4001 is less than 023, then 4001 is cleared to zero and then forced to contain the value 023.

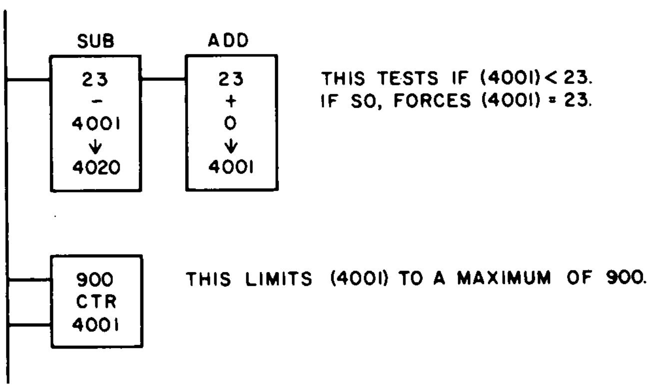


Figure 29. Limiting Register Contents

Keypad Entry of Numerical Data

An alternate method of entering numerical data into the registers of the 484 Controller with an Enhanced instruction set is with a 10-digit numerical keypad. The keypad will be connected to 10 discrete inputs, with each input representing one digit. The logic for this example is shown in Figure 30.

When a number is selected on the keypad, the input will energize which activates the associated ADD function forcing the selected number into a common register, 4101. Also when any number is selected, coil 98 is energized which activates a multiply function which takes the previous contents of register 4101, and multiplies by 10. This function shifts the previously entered number one position to the left, making room for the next number entered.

After three numbers have been entered, the maximum capacity of a register, coil 99, which is driven by a counter with a preset of three, is energized. This signal could be used to either inhibit further entry or drive a register to table move to store the number in a table.

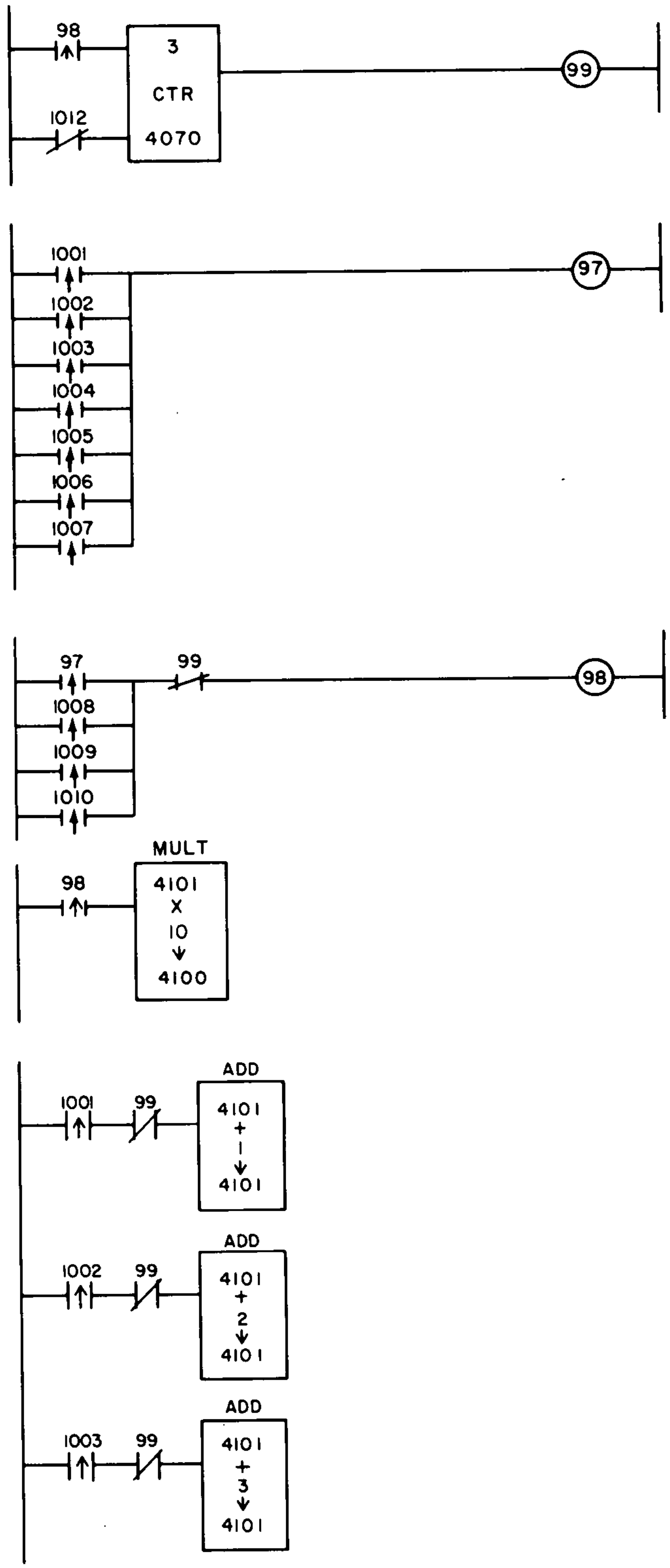


Figure 30. Keypad Data Entry

432 over→

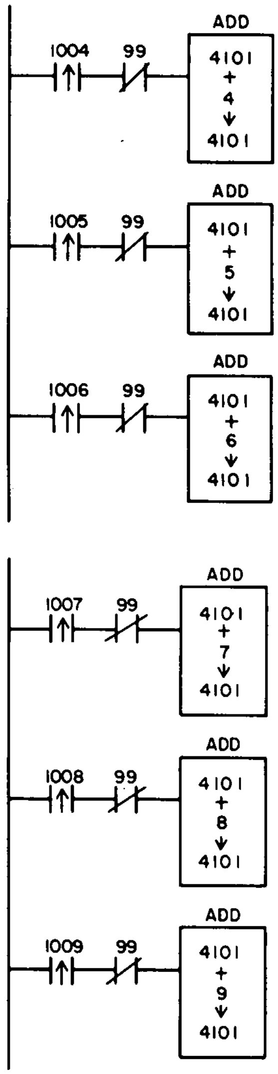


Figure 30. Keypad Data Entry (Cont)

Numerical Shift Register

This program shows an example of a numerical shift register with up to 253 shift registers for a 484 Controller with Enhanced II instruction set. The shift register requires 28 words of memory regardless of shift length. It requires only one discrete signal for shift rather than separate remove and load signals, since removals and loads always occur in pairs. Similarly it requires only one pointer, since the removal/load pair occurs at the same point in the shift register. After each shift, the pointer is incremented. The pointer is reset to the beginning if it points past the end of the shift register.

The example given here uses these addresses:

3001 = input stage
4001 = output stage
4101-4200 = shift register
1001 = shift
4061 = pointer

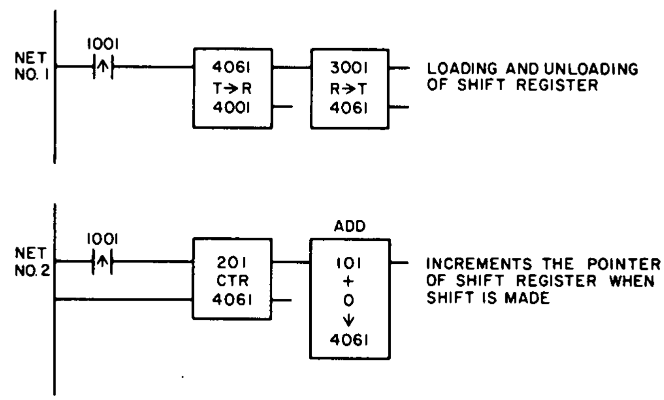


Figure 31. Numerical Shift Register

First In, First Out

This program demonstrates the ability to build a FIFO stack with up to 248 stages. Each stage is a register with a value up to 999. Each stack has stages for loading and removal and uses discrete signals for the load and removal. The example given here used these addresses:

3001 = input stage
4001 = output stage
4101-4200 = FIFO stack
4091 = number of valid entries in FIFO
1001 = load
1002 = remove

Two pointers are used to route loads and removals:

4061 always points to the register into which data will next be loaded.

4071 always points to the register from which data will next be removed.

Each time data is loaded or removed, register 4061 or 4071 is incremented, depending on which action happens. If this register now points to 4201 (i.e., past the end of the FIFO), it is reset to point to 4101 (beginning of FIFO).

433

Two relationships define the status of the FIFO:

1) If 4061 and 4071 point to the same register, the FIFO is empty.
2) If 4071 is exactly one greater than 4061 (allowing for wraparound), the FIFO is full.

The organization of the data is "upside-down" from the arrangement of FIFO data in the 184/384. The next entry will be into the register immediately after the register which holds the last entry.

After entering in the logic shown in Figure 32, initialize the FIFO by setting (4061) = (4071) = 101.

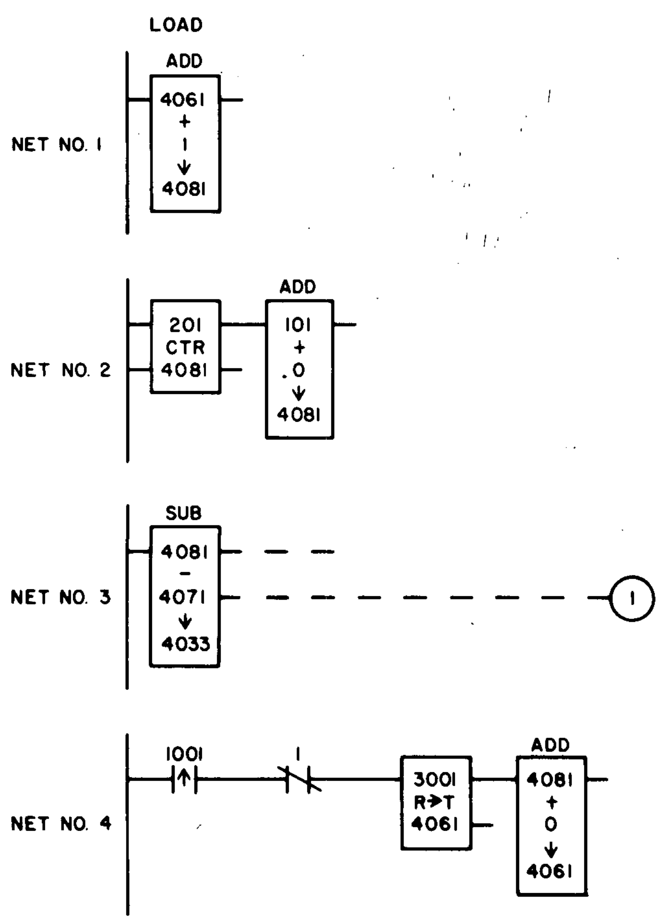


Figure 32. First In – First Out

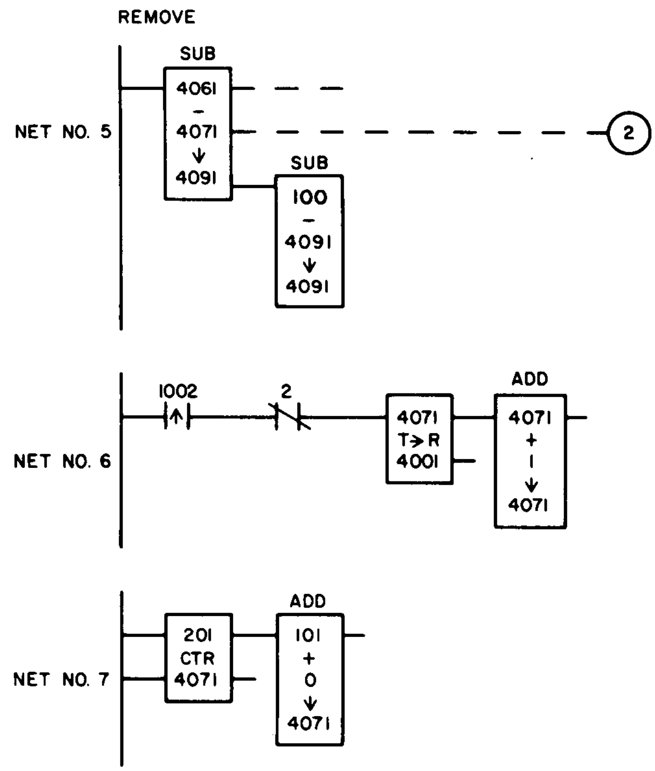


Figure 32. First In – First Out (Cont)

Table Lookup

The table lookup operation will search a table looking for a match to an inputted value and its location in the table.

In this example, shown in Figure 33, the table to be searched will be from register 4101 to 4200, and the inputted data to be matched will be in register 3001. In the first network, upon initiating of the start command, input 1001 forces the value 101 into register 4061 which is the pointer register for the table to register move. This will start the matching operation at register 4101 and then will continue at the rate of one register each scan of the controller until a match is found or until the end of the table is reached.

When a match is found, the locations in the table of the register which is matched will be placed into register 4001 and coil 3 will be energized for one scan.

434 over→

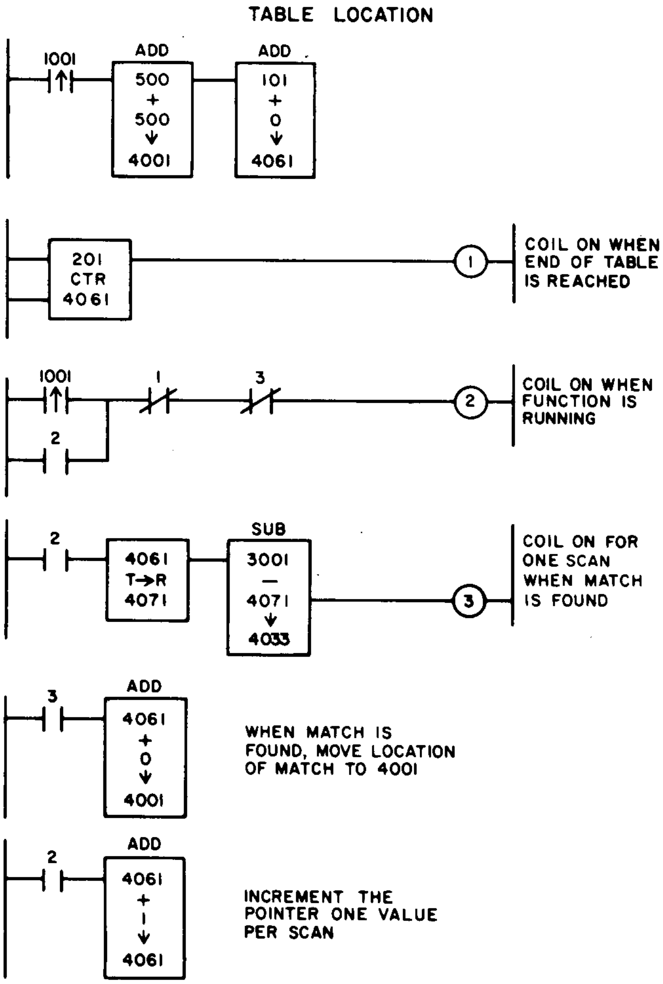


Figure 33. Searching a Table for a Match

Searching a Table for Its Largest Value

In many applications, the 484 Controller with Enhanced II instruction set may be used to gather data from a series of operations and stored in a table of registers. The following logic shown in Figure 34 offers the ability to search the table and find the largest stored value and its location in the table.

The table to be searched in this example will be from register 4101 to 4200, the position of the largest register will be displayed in register 4001 and the largest value will be shown in register 4002.

The first network on initiation of the start search command (input 1001) forces the pointer of the "table to register move" to the second register in the table (4102) to start the compare.

The next network checks to see if the pointer register (4061) equals 201, meaning the search has reached the end of the table. If so, the skip function is actuated stopping any further searching.

The third network performs the comparing operation. The table to register move loads the contents of the table, at one register per scan, into register 4071. That register's contents is compared against the largest value found so far (4002). When found, register 4001 will contain the position in the table of this value and 4002 will contain the largest value.

The last network is a pointer index which every scan adds one to the contents of the pointer register forcing another move from the table to register 4071.

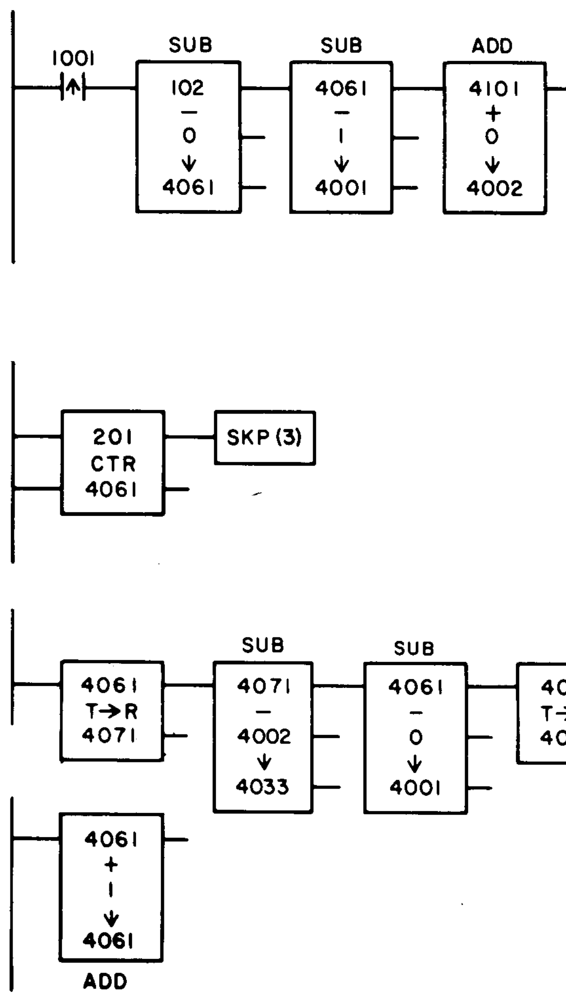


Figure 34. Searching a Table for Its Largest Value

435

Forcing All Registers to Zero

With the Enhanced II instruction set, the contents of all the registers in the controller can be forced to zero when contact 1007 is energized as shown on Figure 35. Only two registers are used for this operation, 4002 with its contents of zero and 4001 which is used as the indexer.

When the control contact 1007 is energized, the ADD function takes the contents of register 4001 and adds the value one to every scan. Register 4001 is used as the pointer for the register to table move, and register 4002 with its contents of zero is the source register. As the pointer is indexed, zero is loaded into successive registers up to the limit of the controller, in the case of a 4K controller 254 registers.

The set point logic will limit the number of registers to set to zero and then reset the pointer to zero.

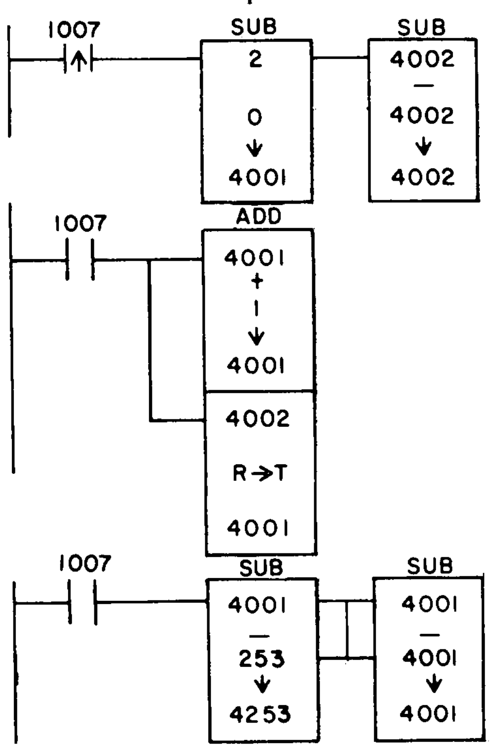


Figure 35. Forcing All Registers to Zero

Sort in Ascending or Descending Order

The following program will perform a sort routine using a 484 Controller with Enhanced II instruction set. This sample routine shown in Figure 36 sorts the table 4101-4110 in either ascending or descending order (switch selectable) and collates the table 4201 accordingly upon closure of 1001. Coil 1 comes on when the sort is finished. If table 4101-4110 was already sorted, coil 3 will come on as well.

This routine will do both ascending and descending sorts, when 1002 is ON, the sort is ascending: if it is OFF, the sort is descending. (Note: it is important that the user does not change the status of this reference while the sort is going on.)

The sort is done not in one scan, but in $n/2(n - 1) + 1$ scans, where n is the number of registers in the table.

When doing an ascending sort, this algorithm functions as follows:

first (n - 1) scans	get smallest element to top of table
next (n - 2) scans	get second-smallest element to second from top
.	.
.	.
.	.
next to last scan	get second-largest element to second position from bottom
last scan	reach end of table

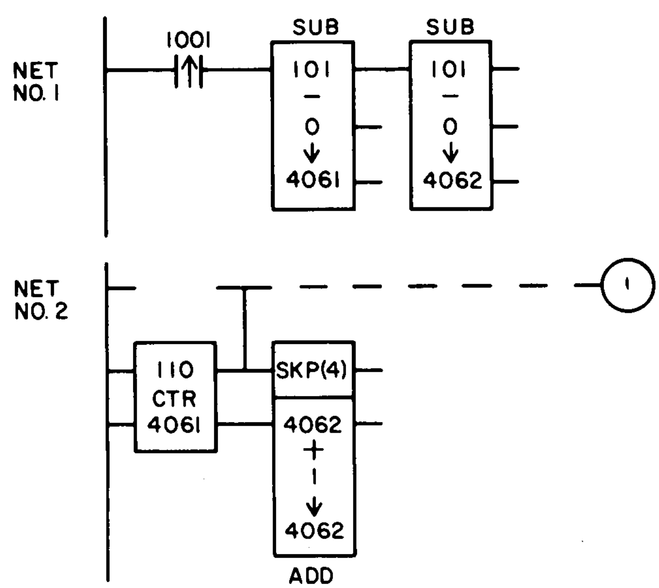


Figure 36. Table Sort

436 over→

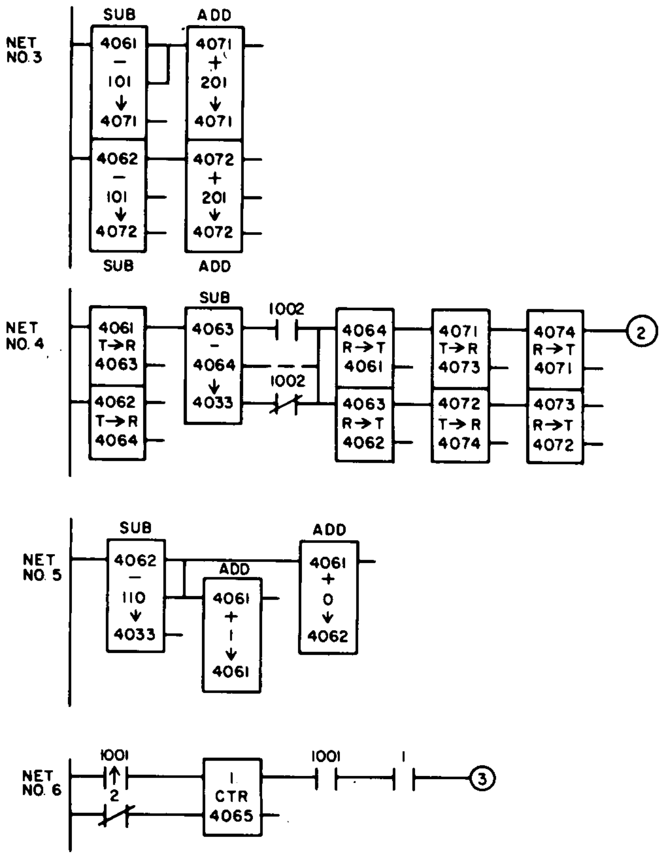


Figure 36. Table Sort (Cont)

Averaging Numerical Values

In some applications, it is desired to average a series of numerical values which are stored in consecutive holding registers. These numerical values could be a series of readings from an analog input module, or the number of parts produced by several machines where the average number is required.

In this example shown in Figure 37, the numerical values to be averaged are stored in registers 4101 through 4105 and the average value found by this logic for the five registers will be stored in register 4004. This logic does not include a start command, so the average value will be updated everytime a value in one of the five registers changes.

The first network will increment the pointer register, 4001, one value every scan of the controller. This pointer will drive the table to register move in network 3.

In the second network, the counter with a preset of 106 will stop the table to register move at register 4105, and will then pass power to the DIVIDE function at the end of the table. The DIVIDE function will take the accumulated total of the table, the contents of registers 4002 and 4003, and divide this value by 5 and deposit the result, which is the average, into register 4004. The MULTIPLY function will clear the contents of registers 4002 and 4003 to zero and the ADDITION function will force the pointer register to 101 to start the table to register move at register 4101.

The third network includes the table to register move with its pointer register 4001 and its destination register of 4005. Next is an accumulating addition function which totals all the contents of the table into register 4003. When the contents of register 4003 exceeds 999, the upper output of the addition function energizes, forcing the next addition function to add one into the contents of register 4002. This provides a double precision addition to allow the accumulated total added from the table to be 1000 or greater before it is averaged.

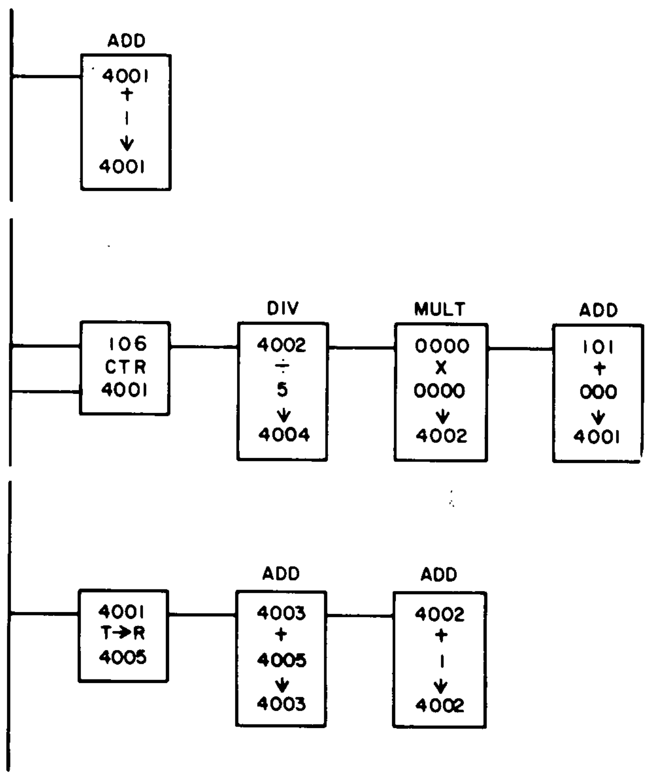


Figure 37. Averaging

Sequencers with More Than 32 Steps

The 484 Controller with the Enhanced instruction set has eight sequencers, each with 32 steps. In the event that a process would require more than 32 steps, several sequencers may be combined, so that the effect would be similar to that of a sequencer with more than 32 steps. The contents of 4051 would control *all* steps in the example below which shows how three sequencers can be combined to act similar to a 96 step sequencer. This technique may be further modified to obtain more than 96 steps, up to the limit of the eight step sequencers.

1001 is the signal to step 1.

1002 may be used as a stop, forcing the sequencer to step 0.

ADD causes the sequencer to jump from step (N+1) to step 1.

(N+1) determines the number of steps, with N being the number of steps wanted.

Contents of 4051	Contents of 4052	Contents of 4053	Typical Effect
0	968	936	All sequencers off
32	0	968	Only 2132 is energized
33	1	969	Only 2201 is energized
34	2	970	Only 2202 is energized
64	32	0	Only 2232 is energized
65	33	1	Only 2301 is energized

To extend a sequence series beyond 96 steps would require one ADD statement per 32 steps or part. The ADD statements would require the following middle values:

Middle Value	Steps	
968	33-64	using 2201-2232
936	65-96	using 2301-2332
904	95-128	using 2401-2432
872	129-160	using 2501-2532
840	161-192	using 2601-2632
808	193-224	using 2701-2732
776	225-256	using 2801-2832

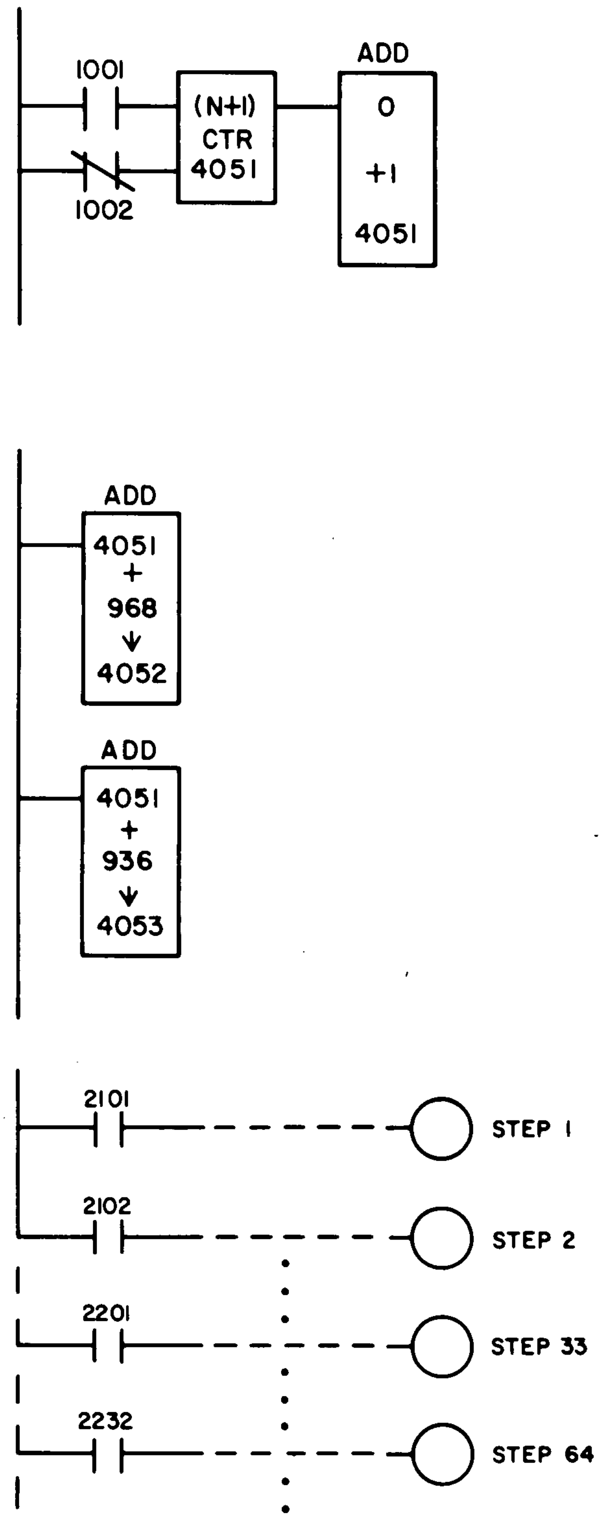


Figure 38. Extended Sequencer

438 over

Binary Input CONVERT

Calculations within the 484 use three digit numbers, maximum value 999. When using a binary input CONVERT which is 10 bit, it is possible to obtain a number whose value is up to 1023. In the event that a particular bit pattern resulting from a binary input CONVERT would have a magnitude > 999, calculates will not yield the results expected. Whenever numbers > 999 are required, it is necessary to use double precision techniques.

The following logic will monitor the result of a binary input CONVERT, and, in the event that the bit pattern would have a magnitude > 999, steer the result to two registers.

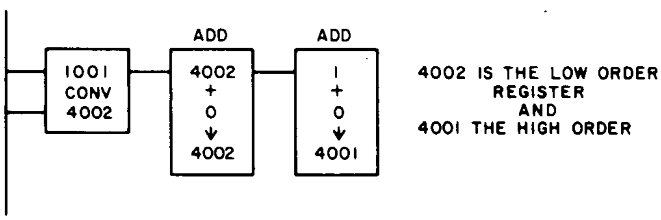


Figure 39. Binary Input Convert

If in the above, only the CONVERT was programmed, register 4002 could contain a bit pattern whose magnitude >999. This bit pattern will be contained in 4002 even though it exceeds 999, because the register can contain the full 10 bits up to 1023 in magnitude. This register should not be used in calculation if its magnitude exceeds 999 because the calculates examine magnitude, and won't accept a number > 999. Timers, counters, R→T and T→R moves do not check magnitude in the same manner, therefore, despite the possibility of register 4002 containing a number whose magnitude > 999, it could still be used as the preset for a Timer or Counter, or moved using T→R and R→T moves. Its contents cannot, however, be examined by using the P180.

Gray Code to BCD Conversion

The 484 Controller with Enhanced II instruction set can readily convert binary- or BCD-coded device inputs to numerical values by means of the BCD CONVERT function. For example, if 1001-1012 represent a binary-coded series of numerical inputs, this logic will place that numerical value in register 4001.

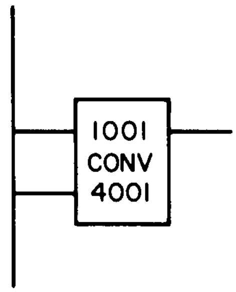


Figure 40

If 1001-1012 represent a BCD-coded series of inputs, this logic places the numerical value in 4001:

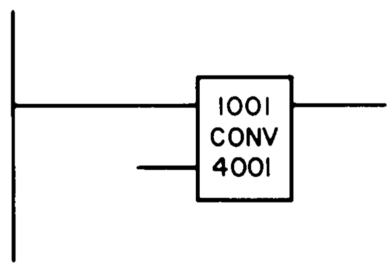


Figure 41

There exists a third common coding method, the Gray code, for which there is no direct CONVERT function on the 484. Herewith is presented a user program to convert Gray code inputs to numerical values.

The general algorithm for deriving the binary equivalent (B) of a Gray code bit pattern (G) is as follows:

Starting conditions: B = all zeroes
start at the LSB of G

439

1) If this bit in G is a "1", complement all the bits in B from the present bit to the LSB. If not, do nothing.
2) If there are no "1" bits in G to the left of the present bit, stop – you are done.
3) Move one bit to the left.
4) Go to step 1.

Example: Find the binary bit pattern which is equivalent to 101011 (Gray code).

G	B
101011	000000
1	1
1	10
0	010
1	1101
0	01101
1	110010 is the solution

To complement an n-bit binary number, simply subtract it from $2^n - 1$.

Example: Find the complement of 001101.

$$\begin{array}{r} 111111 \\ -\ 001101 \\ \hline 110010 \end{array}$$

The 484 can perform these subtractions easily enough. Even if $2^n - 1 > 999$, the 484 can still do double precision subtractions (networks 11-17). This program will convert up to 16 bits of Gray code (inputs 1001-1016) to a double-precision BCD number (registers 4001-4002). If fewer than 16 bits are used, some of the discrete inputs and logic are omitted. For example, with 12 bits of Gray code, inputs 1001-1004 and the networks which refer to them are not used. If the Gray code data consists of nine or fewer bits, all logic which refers to register 4001 can be omitted. The BCD value will simply be returned (single-precision) in 4002.

Memory required to implement this conversion is as follows:

Number of Gray Code Bits	Words of Memory Required
1	18
2	28
3	38
4	48
5	58
6	68
7	78
8	88
9	98
10	128
11	158
12	188
13	218
14	248
15	278
16	308

The utility of this program is subject to a few restrictions imposed by the 484 scanning mechanism. Consider an n-bit Gray-coded shaft encoder attached to a shaft which rotates at a speed of R rev/min. If the scan time of the 484 is T msec, the value of 4001-4002 may differ from the actual shaft position by as much as:

$$\text{error} = \frac{2^n\,RT}{60{,}000}\ \text{steps}$$

Example: A 10-bit encoder is connected to a shaft which rotates at 25 RPM. The scan time of the 484 is 20 msec.

$$\text{The maximum error is } \frac{2^{10} \times 25 \times 20}{60{,}000} = 8 \text{ or } 9 \text{ steps.}$$

Since one rotation consists of 1024 steps, this error is 0.8 or 0.9%.

4/10 over

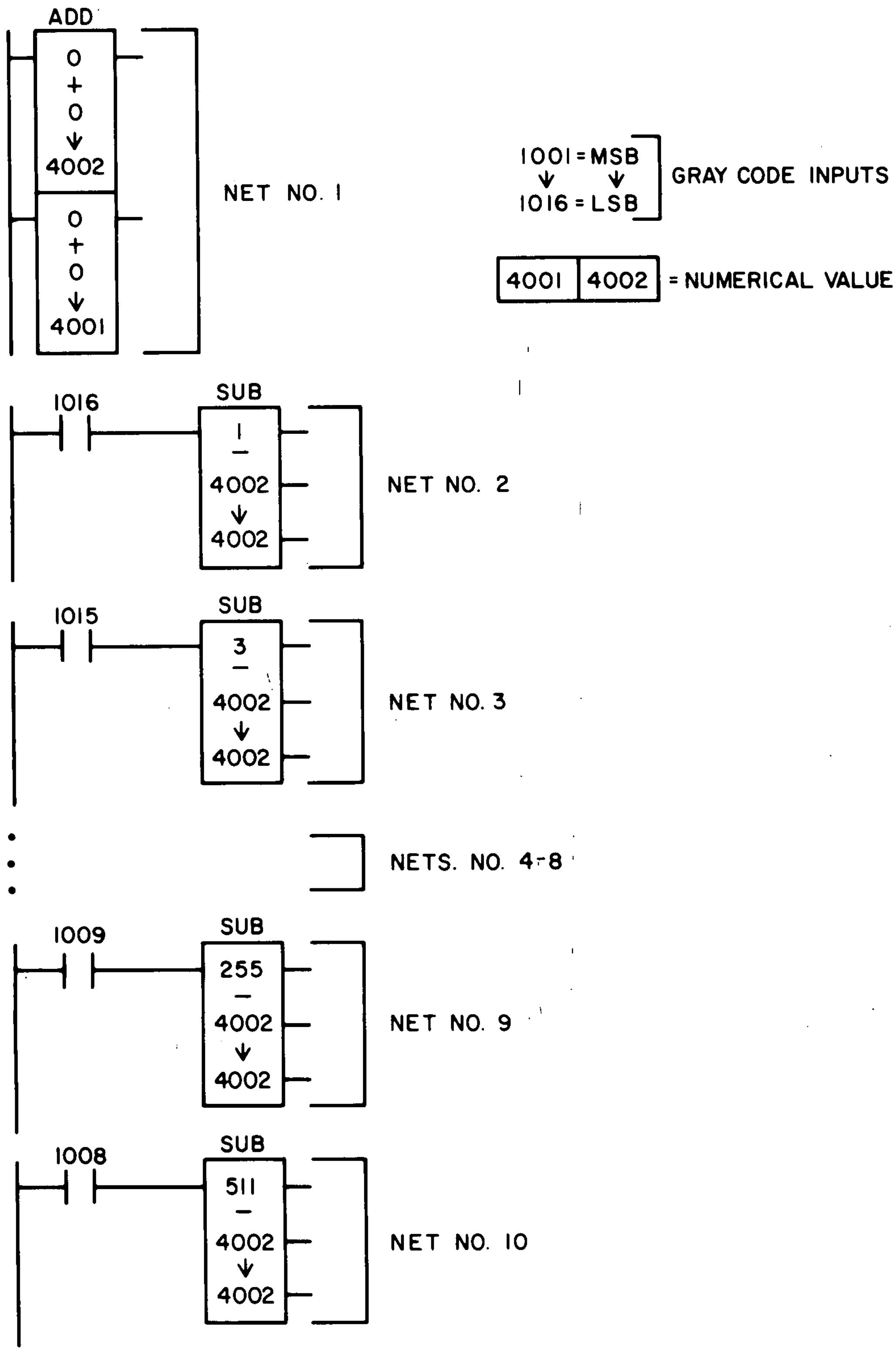


Figure 42. 484 Gray Code to BCD Conversion

441

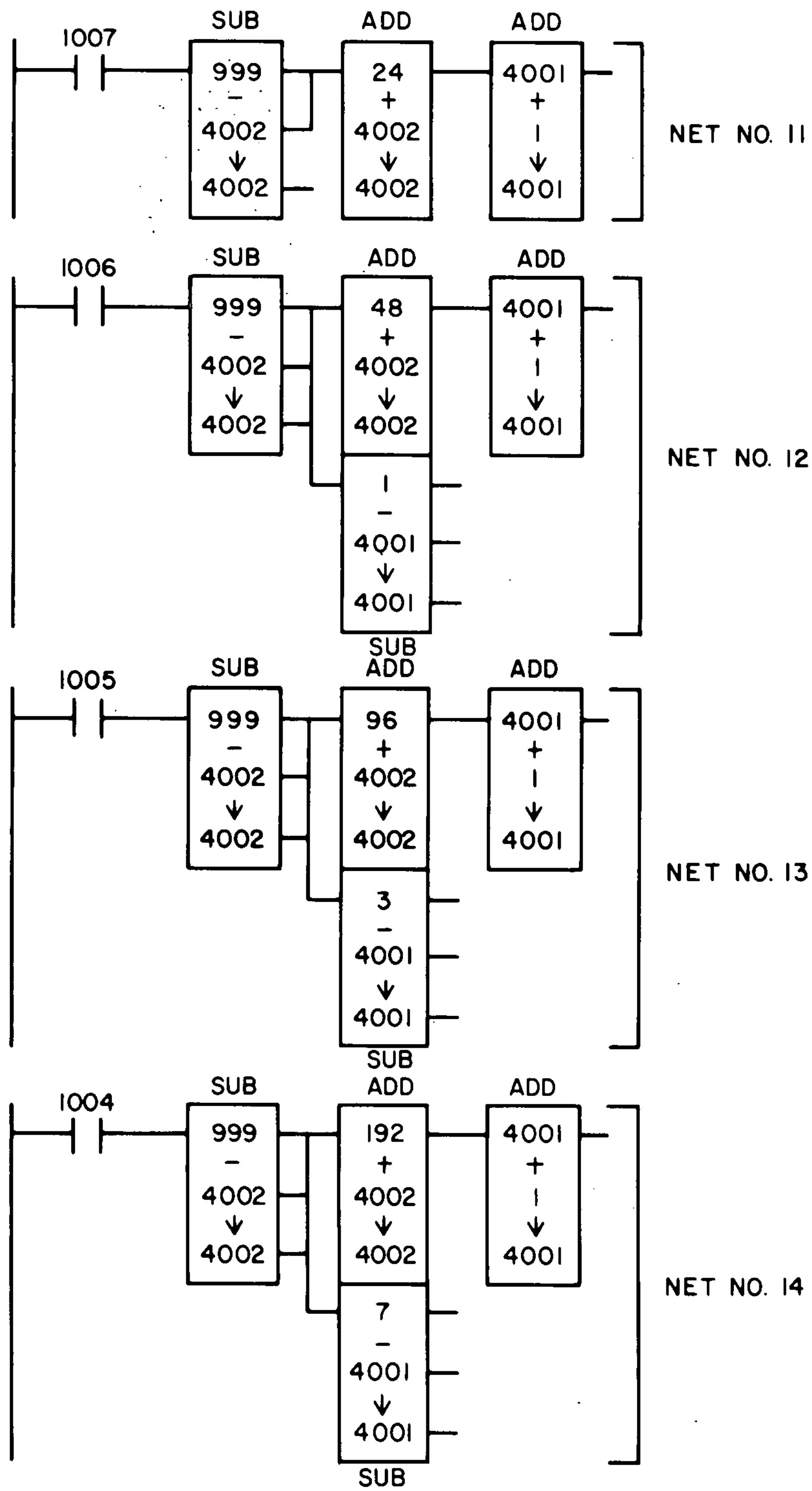


Figure 42. 484 Gray Code to BCD Conversion (Cont)

442 over →

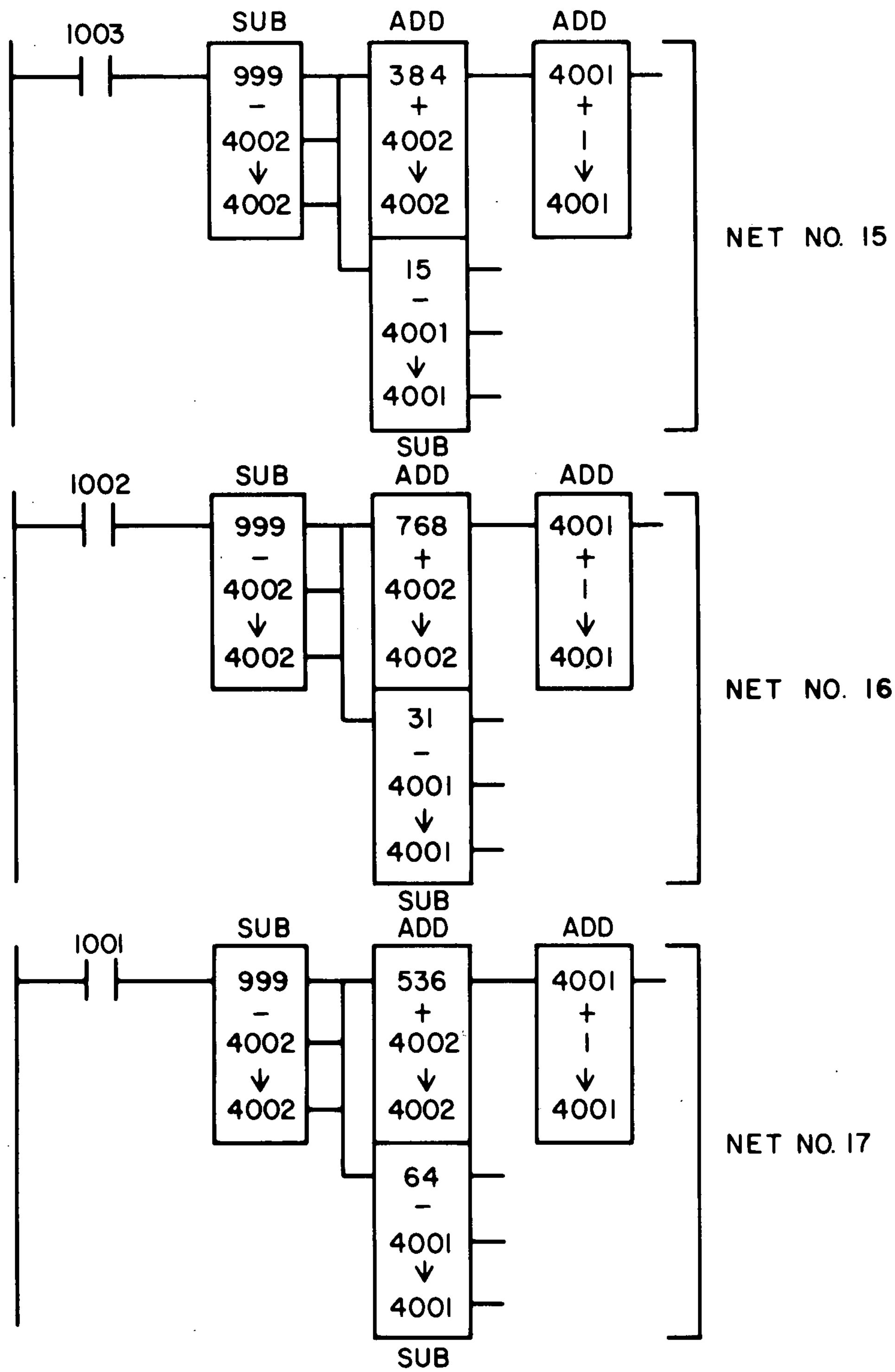


Figure 42. 484 Gray Code to BCD Conversion (Cont)

443

EXHIBIT F

5th

pH Control

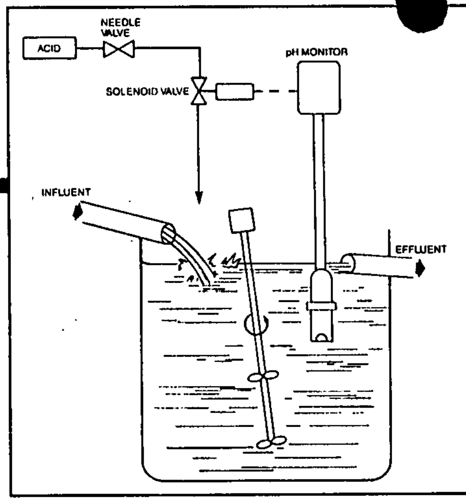


Figure 2. Continuous control of pH involves constant flow of solution in and out of treatment tank.

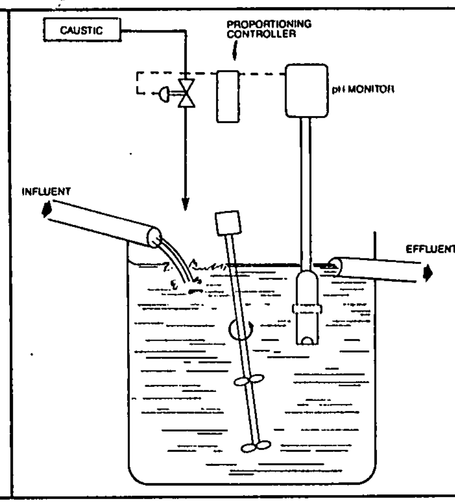


Figure 3. Proportioning controller is used to adjust pH with caustic on this system where influent's pH fluctuates greatly. Throttling valve automatically regulates reagent flow.

Accurate measurement and control of pH is a common requirement in industrial plants. Applications include monitoring of cooling tower water, boiler feedwater, process steam, waste treatment, and plant effluent. Control of pH is dependent on measurement reliability, and proper measurement requires an understanding of the basic principles.

By definition, pH is the logarithm of the expression 1/(hydrogen-ion concentration). The concentration is expressed in moles per liter.

The linear scale for pH is not linear with concentration because it is a logarithmic relationship. A change of one pH unit represents a tenfold change in the effective strength of an acid or base.

Control Factors—Normally, one of two conditions exists when controlling pH: either the solution is too acidic and a base must be added to reach a specified higher pH, or the solution is too alkaline and an acid must be added to lower the pH. In both cases, the corrective medium (called a reagent) must be added at a controlled rate.

A key objective in designing pH control systems is to minimize the amount of required reagent. But determining and feeding the exact amount can be difficult because of the logarithmic relationship. Overshooting pH limits by adding more than the correct amount of reagent is easy to do if the control system is improperly designed.

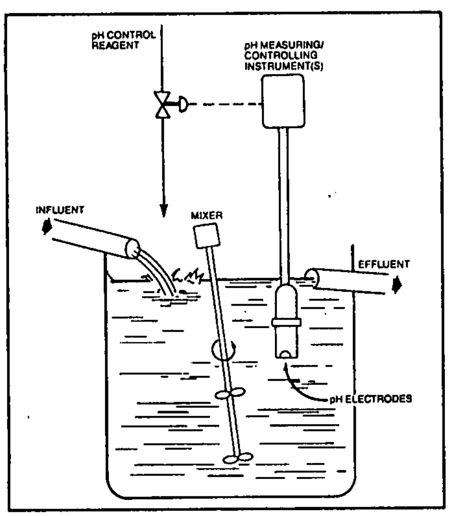


Figure 1. Batch control of pH is normally used when solution volumes are relatively small. Efficient mixing and proper location of pH electrodes is critical to ensure accurate results.

Control Techniques—Basically, pH control involves supplying the proper amount of reagent to bring the pH to the desired value. Control can be performed on either a batch or continuous basis.

Batch control is normally used when the total volume of the solution to be treated is relatively low, such as in waste treatment processes where liquids can be collected efficiently and treated in tanks. The amount of reagent required for neutralization can be determined from a titration curve, tank volume, and reaction time. Slow reagent addition rates and good stirring permit more accurate control with less likelihood of pH overshooting.

In the batch process shown in Fig. 1, the tank inlet and reagent feed point are shown located away from the pH electrodes and effluent discharge pipe. This separation is necessary to ensure proper mixing of reagent before measurements are made. Immersed electrodes near the outlet ensure rapid sensor response.

Continuous control is similar to batch control except that there is a continuous flow of influent and treated effluent, Fig. 2. A proportional controller may be required to regulate reagent flow rates if influent pH varies widely, Fig. 3.

Electrode Assembly Placement—An improperly placed electrode assembly can cause excessive deadtime for control action and result in cyclic control and wasted reagent. Deadtime is defined as the elapsed time between reagent addition and the first measureable pH change resulting from the addition. Ideal deadtimes range from 5 to 30 sec. Excessive deadtime can often be avoided by locating the electrode assembly close to where the reagent is added.

Vertical mounting of electrodes is preferred, and they should always be exposed to a representative sample of the process solution. The entire assembly should remain wet at all times to keep the electrodes from drying out.

Because pH is a high-impedance measurement, it is best to utilize a controller, preamplifier or signal conditioner as close to the electrode as possible. The preamplifier converts a high-impedance signal to a low-impedance signal, making it less susceptible to noise and signal loss on transmission back to the receiving instrument over unshielded wire.

Maintenance—A regularly scheduled maintenance program must be enacted to keep electrodes clean and calibrated. Electrode assemblies should be equipped with an ultrasonic cleaning device if solutions contain high levels of suspended solids that are fibrous or crystalline in nature.

Some solutions will chemically attack the electrodes and/or electrode housing material. For example, at elevated temperatures, highly alkaline (pH above 12) solutions can damage glass electrodes or cause significant sodium-ion errors. Fluoride solutions with a pH below 4 will quickly dissolve glass membranes. Process solutions above 230°F will significantly reduce electrode life. Maximum life is normally achieved at ambient temperatures. Coolers are recommended for extremely hot sampling situations.

Based on the article "Understanding pH Measurement and Control" from PLANT ENGINEERING Magazine. Used with permission.

517

Thermocouple Reference Junction Principles

THEORY: When accurate thermocouple measurements are required, it is common practice to reference both legs to copper lead wire at the ice point so that copper leads may be connected to the emf readout instrument. This procedure avoids the generation of thermal emfs at the terminals of the readout instrument. Changes in reference junction temperature influence the output signal and practical instruments must be provided with a means to cancel this potential source of error. The EMF generated is dependent on a difference in temperature, so in order to make a measurement the reference must be known. This is shown schematically in Fig. #1 and can be accomplished by placing the reference junction in an ice water bath at a constant 0°C (32°F). Because ice baths are often inconvenient to maintain and not always practical, several alternate methods are often employed.

ELECTRICAL BRIDGE METHOD: This method usually employs a self-compensating electrical bridge network as shown in Figure 2. This system incorporates a temperature sensitive resistance element (R_T), which is in one leg of the bridge network and thermally integrated with the cold junction (T_2). The bridge is usually energized from a mercury battery or stable d.c. power source. The output voltage is proportional to the unbalance created between the pre-set equivalent reference temperature at (T_2) and the hot junction (T_1). In this system, the reference temperature of 0°C or 32°F may be chosen.

As the ambient temperature surrounding the cold junction (T_2) varies, a thermally generated voltage appears and produces an error in the output. However, an automatic equal and opposite voltage is introduced in series with the thermal error. This cancels the error and maintains the equivalent reference junction temperature over a wide ambient temperature range with a high degree of accuracy. By integrating copper leads with the cold junction, the thermocouple material itself is not connected to the output terminal of the measurement device, thereby eliminating secondary errors.

THERMOELECTRIC REFRIGERATION METHOD: The Omega TRC Thermoelectric Ice Point Reference Chamber relies on the actual equilibrium of ice and distilled, deionized water and atmospheric pressure to maintain several reference wells at precisely 0°C. The wells are extended into a sealed cylindrical chamber containing pure distilled, deionized water. The chamber outer walls are cooled by thermoelectric cooling elements to cause freezing of the water in the cell. The increase in volume produced by freezing an ice shell on the cell wall is sensed by the expansion of a bellows which operates a microswitch, de-energizing the cooling element. The alternate freezing and thawing of the ice shell accurately maintains a 0°C environment around the reference wells. An application schematic is shown in Fig. #3.

Completely automatic operation eliminates the need for frequent attention required of common ice baths. Thermocouple readings may be made directly from ice point reference tables, such as those listed in the technical section, without making corrections for reference junction temperature. Any combination of thermocouples may be used with this instrument by simply inserting the reference junctions in the reference wells. Calibration of other type temperature sensors at 0°C may be performed as well.

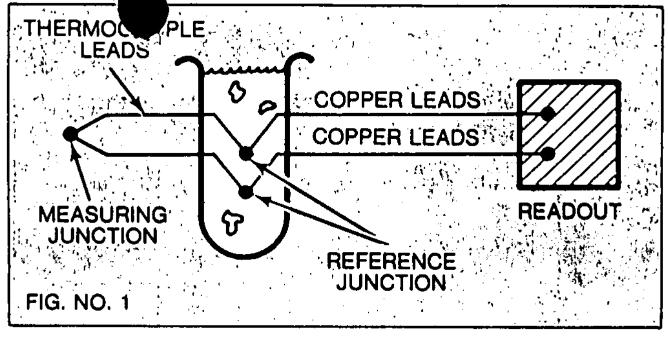


FIG. NO. 1

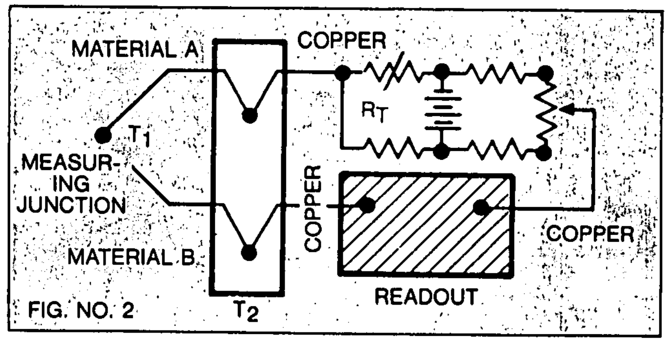


FIG. NO. 2

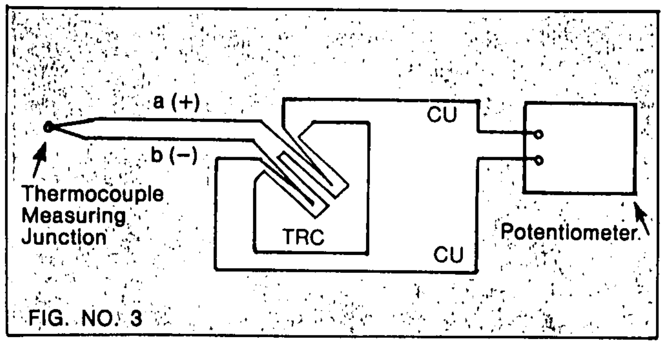


FIG. NO. 3

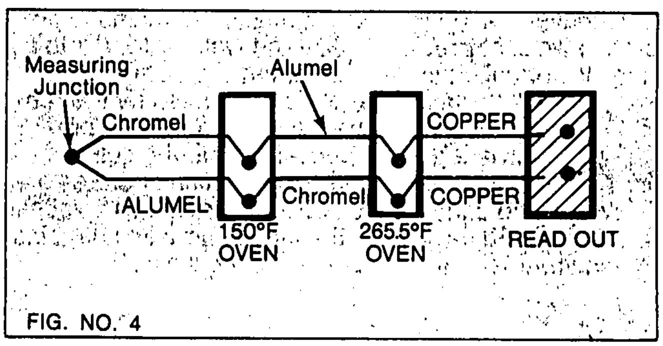


FIG. NO. 4

HEATED OVEN REFERENCES:
The double-oven type employs two temperature-controlled ovens to simulate ice-point reference temperatures as shown in Fig. 4. Two ovens are used at different temperatures to give the equivalent of a low reference temperature differing from the temperature of either oven. For example, leads from a Chromel-Alumel thermocouple probe are connected with a 150° oven to produce a Chromel-Alumel and an Alumel-Chromel junction at 150°F (2.66 mv each).

The voltage between the output wires of the first oven will be twice 2.66 mv or 5.32 mv. To compensate for this voltage level, the output leads (Chromel and Alumel) are connected to copper leads within a second oven maintained at 265.5°F. This is the precise temperature at which Chromel-Copper and Alumel-Copper produce a bucking voltage differential of 5.32 mv. Thus, this voltage cancels out the 5.32 mv differential from the first oven leaving 0 mv at the Copper output terminals. This is the voltage equivalent of 32°F (0°C).

Introduction to pH

INTRODUCTION

pH is a unit of measure which describes the degree of acidity or alkalinity of a solution. It is measured on a scale of 0 to 14. The term pH is derived from "p", the mathematical symbol of the negative logarithm, and "H", the chemical symbol of Hydrogen. The formal definition of pH is the negative logarithm of the Hydrogen ion activity.

$$pH = -\log[H+]$$

pH provides the needed quantitative information by expressing the degree of the activity of an acid or base in terms of hydrogen ion activity.

The pH value of a substance is directly related to the ratio of the hydrogen ion $[H^+]$ and the hydroxyl ion $[OH^-]$ concentrations. If the H^+ concentration is greater than OH^-, the material is acidic; i.e., the pH value is less than 7. If the OH^- concentration is greater than H^+, the material is basic, with a pH value greater than 7. If equal amounts of H^+ and OH^- ions are present, the material is neutral, with a pH of 7.

Acids and bases have free hydrogen and hydroxyl ions, respectively. Since the relationship between hydrogen ions and hydroxyl ions in a given solution is constant for a given set of conditions, either one can be determined by knowing the other. Thus, pH is a measurement of both acidity and alkalinity, even though by definition it is a selective measurement of hydrogen ion activity. Since pH is a logarithmic function, a change of one pH unit represents a ten-fold change in hydrogen ion concentration. Table 1 shows the concentration of both the hydrogen ion and the hydroxyl ion at different pH values.

HYDROGEN ION [illegible] IN MO[illegible]		
pH	**H+**	**OH−**
0	(10^{0}) 1	0.00000000000001 (10^{-14})
1	(10^{-1}) 0.1	0.0000000000001 (10^{-13})
2	(10^{-2}) 0.01	0.000000000001 (10^{-12})
3	(10^{-3}) 0.001	0.00000000001 (10^{-11})
4	(10^{-4}) 0.0001	0.0000000001 (10^{-10})
5	(10^{-5}) 0.00001	0.000000001 (10^{-9})
6	(10^{-6}) 0.000001	0.00000001 (10^{-8})
7	(10^{-7}) 0.0000001	0.0000001 (10^{-7})
8	(10^{-8}) 0.00000001	0.000001 (10^{-6})
9	(10^{-9}) 0.000000001	0.00001 (10^{-5})
10	(10^{-10}) 0.0000000001	0.0001 (10^{-4})
11	(10^{-11}) 0.00000000001	0.001 (10^{-3})
12	(10^{-12}) 0.000000000001	0.01 (10^{-2})
13	(10^{-13}) 0.0000000000001	0.1 (10^{-1})
14	(10^{-14}) 0.00000000000001	1 (10^{0})

Table 1

THE MOLAR CONCEPT

A mole of a compound is defined as Avogadro's number of molecules (6.02×10^{23} molecules), which has a mass approximately equal to the molecular weight, expressed in grams. For example, sodium hydroxide, NaOH, which has a molecular weight of 23 + 16 + 1 = 40, would have 40 grams in a mole. Since the atomic weight of the hydrogen ion (H^+) is one (1), there is one gram of hydrogen ions in a mole of hydrogen. A solution with a pH of 10 has 1×10^{-10} moles of hydrogen ions, or 10^{-10} grams in a one liter solution.

IONIZATION

An ion is a charged particle, created by an atom or molecule which has either gained or lost electron(s). The presence of ions in solution allows electrical energy to be passed through the solution as a conductor. Different compounds form ions in solution in different amounts, depending on the ability of the atoms to gain or lose electrons. They will dissociate (or ionize) in solution to form hydrogen (H^+) or hydroxyl (OH^-) ions in the solution.

Molecules that dissociate easily will form strong acids or bases when in aqueous solution (water solvent). Examples of these are hydrochloric acid (HCl) or sodium hydroxide (NaOH):

$$HCl + H_20 \rightarrow H_30^+ + Cl^-$$
$$NaOH \rightarrow Na^+ + OH^-$$

In an aqueous solution, hydrogen ions normally combine with the water solvent to form the hydronium ion (H_30^+). pH measurements of these solutions are therefore measurements of the hydronium ion concentration. Normally, the terms "hydronium ion" and "hydrogen ion" are used interchangeably in pH measurement applications.

Some compounds form weak acids or bases; only a very small percentage of the compounds dissociates into its constituent ions, so very few hydrogen or hydroxyl ions are formed. An example of this is acetic acid, which forms less than one hydrogen ion for every one hundred molecules:

$$H_20 + CH_3COOH \rightarrow H_30^+ + CH_3COO^-$$

Pure water also dissociates weakly, with 10^{-7} hydrogen and 10^{-7} hydroxyl ions formed for every water molecule at 25°C:

$$2H_20 \rightarrow H_30^+ + OH^-$$

The addition of acid to water increases the concentration of hydrogen ions and reduces the concentration of hydroxyl ions. A base added to water has the opposite effect, increasing the concentration of hydroxyl ions and reducing the concentration of hydrogen ions:

$$H_20 + HCl \rightarrow H_30^+ + Cl^-$$
$$H_20 + NaOH \rightarrow Na^+ + H_20 + OH^-$$

There is a wide variety of applications for pH measurement. For example, pH measurement and control is the key to the successful purification of drinking water, the manufacture of sugar, sewage treatment, food processing, electroplating, and the effectiveness and safety of medicines, cosmetics, etc. Plants require the soil to be within a certain pH range in order to grow properly, and animals can sicken or die if their blood pH level is not within the correct limits. Figure 1 gives pH values for some common industrial and household products.

pH MEASUREMENT

A rough indication of pH can be obtained using pH papers or indicators, which change color as the pH level varies. These indicators have limitations on their accuracy, and can be difficult to interpret correctly in colored or murky samples.

More accurate pH measurements are obtained with a pH meter. A pH measurement system consists of three parts: a pH measuring electrode, a reference electrode, and a high input impedance meter. The pH electrode can be thought of as a battery, with a voltage that varies with the pH of the measured solution. The pH measuring electrode is a hydrogen ion sensitive glass bulb, with a millivolt output that varies with the changes in the relative hydrogen ion concentration inside and outside of the

513

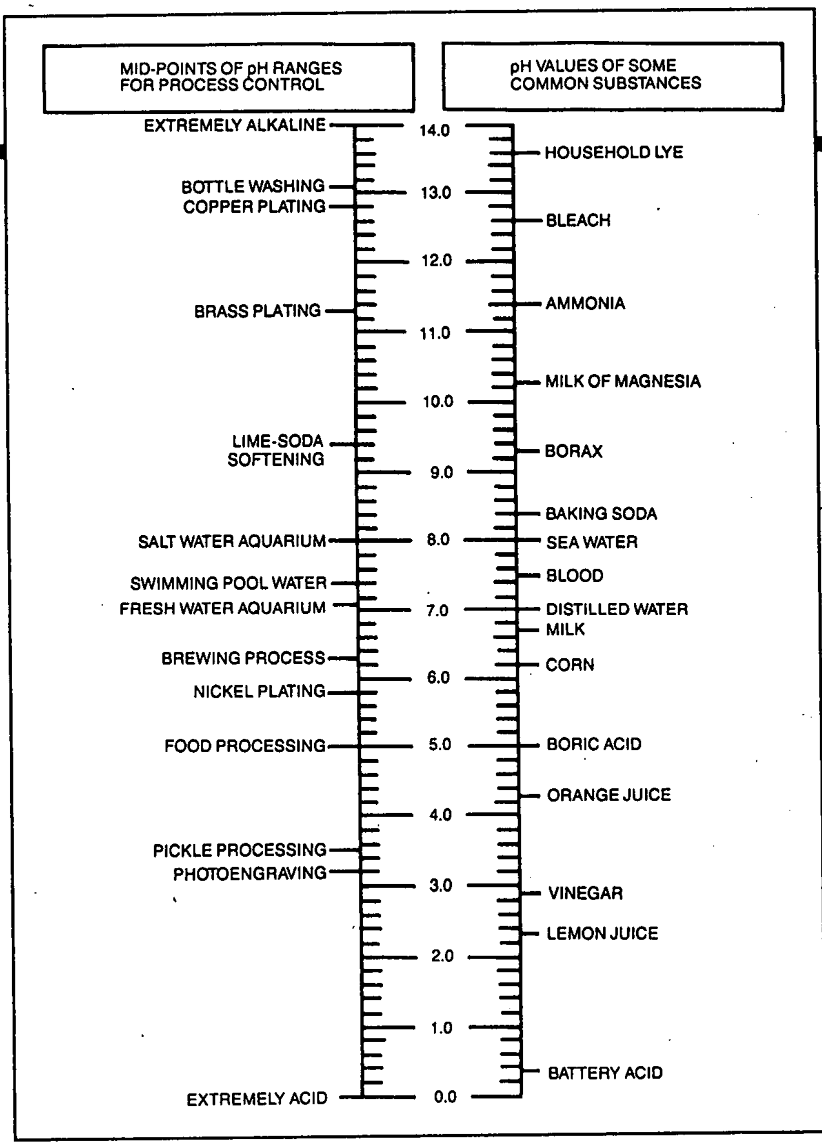


Figure 1

bulb. The reference electrode output does not vary with the activity of the hydrogen ion. The pH electrode has very high internal resistance, making the voltage change with pH difficult to measure. The input impedance of the pH meter and leakage resistances are therefore important factors. The pH meter is basically a high impedance amplifier that accurately measures the minute electrode voltages and displays the results directly in pH units on either an analog or digital display. In some cases, voltages can also be read for special applications or for use with ion-select or Oxidation-Reduction Potential (ORP) electrodes.

TEMPERATURE COMPENSATION

Temperature compensation is contained within the instrument, because pH electrodes and measurements are temperature sensitive. The temperature compensation may be either manual or automatic. With manual compensation, a separate temperature measurement is required, and the pH meter manual compensation control can be set with the approximate temperature value. With automatic temperature compensation (ATC), the signal from a separate temperature probe is fed into the pH meter, so that it can accurately determine pH value of the sample at that temperature.

BUFFER SOLUTIONS

Buffers are solutions that have constant pH values and the ability to resist changes in that pH level. They are used to calibrate the pH measurement system (electrode and meter). There can be small differences between the output of one electrode and another, as well as changes in the output of electrodes over time. Therefore, the system must be periodically calibrated. Buffers are available with a wide range of pH values, and they come in both premixed liquid form or as convenient dry powder capsules. Most pH meters require calibration at several specific pH values. One calibration is usually performed near the isopotential point (the signal produced by an electrode at pH 7 is 0 mV at 25°C), and a second is typically performed at either pH 4 or pH 10. It is best to select a buffer as close as possible to the actual pH value of the sample to be measured.

TEMPERATURE EFFECTS

As previously stated, the pH electrode is temperature dependent, and may be compensated for in the pH meter circuitry. The circuitry of the pH meter utilizes the Nernst equation, which is a general mathematical description of electrode behavior.

$$E = E_x + \frac{2.3RT_K}{nF} \log(a_i)$$

where:

E_x = constant depending upon reference electrode
R = constant
T_K = absolute temperature (Kelvin)
n = charge of the ion (including sign)
F = constant
a_i = activity of the ion

For pH measurement, we are interested in the hydrogen ion for H^+:

$$\frac{2.3RT_K}{nF} = 59.16 \text{ mV}$$

where: $n = 1$ and $T = 25°C$. This term is commonly known as the Nernst coefficient. Since pH is defined as the negative logarithm of the hydrogen ion activity, the general equation at any temperature can be expressed as:

$$E = E_x - 1.98\, T_K\, \text{pH}$$

Changes in temperature of a solution will vary the millivolt output of the glass pH electrode in accordance with the Nernst equation. Its variation in the electrode sensitivity versus temperature is a linear function, and most pH meters have circuitry designed to compensate for this effect (refer to *Temperature Compensation*). Figure 2 shows the effect on the glass pH electrode signal at various temperatures.

In figure 2, all three slopes intersect at the point of 0 mV and pH 7.0; this implies no millivolt change with temperature at this, the isopotential point. Also, it can be seen that when working near 7.0 pH, temperature compensation is not a significant factor. However, when working at pH levels of 3.0 or 11.0, a temperature change of 15°C can result in an error of 0.2 pH. Since the temperature effect on the electrode has been shown to be linear, the temperature dependence of pH can then be expressed as:

0.03 pH error/pH unit/10°C

The actual pH of the sample can change with temperature due to a change in the hydrogen ion activity in the solution, because ionization of compounds and

514

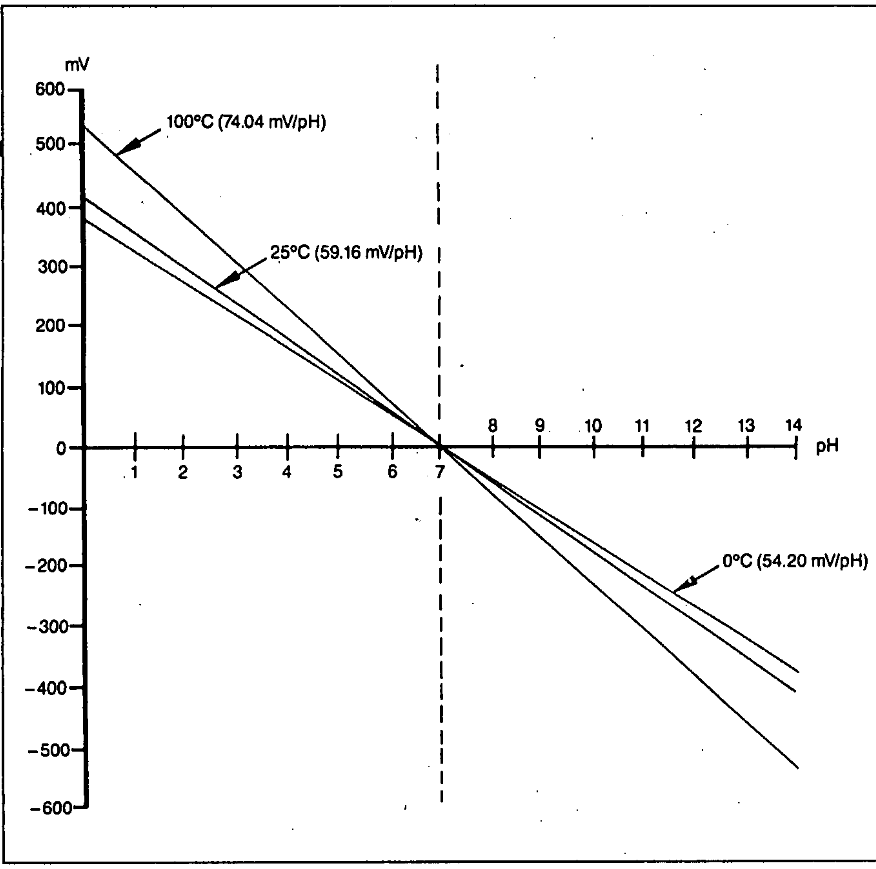


Figure 2

hydrogen ion activity in the solution may be temperature dependent. Temperature compensation does not correct for this, and is not desirable, because an accurate pH measurement is desired at that particular temperature. Temperature compensation only corrects for the change in the output of the electrode, not for the change in the actual solution pH.

Temperature will also affect the glass membrane's impedance. For each 8° below 25°C, the specified impedance approximately doubles. Depending on the original impedance of the glass membrane, the meter will have to handle a higher impedance at a lower temperature.

Written by: OMEGA ENGINEERING, INC.

Measurement of pH

The measurement of pH is one of the most common analytical techniques used in chemistry laboratories today. Nonetheless, pH measurements often suffer from the effects of incorrect materials or incorrect maintenance. The purpose of this paper is to help simplify the choice of materials, methods, and maintenance protocols for pH measurements.

pH MEASUREMENT SYSTEM

A pH measurement system always consists of four parts: a pH sensing electrode, an amplifier that translates the signal into something the user can read, a reference electrode, and the sample being measured. Each part of the system plays a critical role in the measurement process.

A glass electrode is actually a small battery (technically, a transducer). This battery displays a varying voltage, depending upon the pH of the solution in which it is immersed. A reasonable representation of that voltage is given in Figure 1. The potential of the glass electrode is a function of the activity of the free-hydrogen ions and a value E_0 which is supposed to be the 0 or rest potential of the system. This is actually the voltage of the system when the pH is 0.

The reference electrode is also a battery; however, unlike the pH electrode, its voltage does not vary with the activity of hydrogen ion or any other ion solution, but is a function only of the value E_0 or the rest potential.

There are nine actual voltages in the system. Inside the body of the electrode is a wire — normally a silver wire coated with silver chloride. On that wire is some matrix that should present a constant voltage to the wire. At the interface of the wire and solution is a voltage which we can call E_1. Between the wire and the inner surface of the glass is another voltage, on the inner surface of the glass is a voltage (E_3), and across the glass membrane there is a voltage (E_4), which is called the asymmetry potential. There is a voltage on the outer surface of the electrode, a voltage between the pH electrode and the reference electrode; another voltage, referred to as the liquid junction potential or streaming potential, at the point where the filling solution of the reference electrode contacts the sample (E_7); a voltage between the inner surface of the reference electrode and the metal wire that connects the inner filling solution of the reference electrode to the lead wire, and, of course, another voltage on the surface of the connecting wire (E_9).

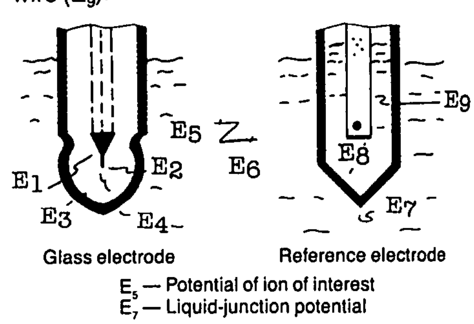


Figure 1. *Flow chart of selection process for reference electrodes.*

When making a pH measurement, one assumes that all of those voltages remain constant except the voltage on the outer surface of the pH electrode. If that assumption is correct, legitimate pH measurements can be made; if not, then incorrect pH measurements will certainly result. In other words, it is

assumed that the pH electrode delivers a varying voltage to the pH meter, while the reference electrode delivers a constant voltage to the meter.

THE REFERENCE ELECTRODE

The reference electrode is the most complicated part of a pH measurement system. When application problems arise, they normally devolve to the reference electrode. When difficulties in pH measurements are encountered, the source of the problem is typically the reference electrode. In our experience the reference electrode accounts for 70% of the problems that arise in the pH measuring process.

A reference electrode consists of three principal parts: an internal element, which is normally either a silver wire coated with silver chloride or a platinum wire covered with a mixture of calomel (Hg_2Cl_2), some filling solution, and a permeable junction through which the filling solution escapes the electrode (called the fluid junction or liquid junction). The liquid junction can come in several forms, but its principal function is to allow small quantities of the reference electrode's filling solution to slowly leak or migrate into the sample being measured. There are three common forms of this junction: 1) ceramic or other frit material, 2) a fibrous material (the best of which is quartz fiber) or 3) a sleeve junction.

Fritted materials are usually white and are composed of small particles pressed closely together. The filling solution leaks through the open cells between these particles. Since the cells vary considerably in size, the flow rate across the surface is quite variable. In some areas, there is almost no net outward flow of filling solution. At those points the junction is considered to be a "diffusion junction." In the areas where the net outward migration of filling solution is rapid, the electrode is considered to have a flowing junction.

Fiber references come in two types: woven fibers and straight fibers. Woven fibers, such as asbestos fiber, have cells of varying size because of the structure of the woven material. Some of these cells are very small and tightly packed together; at those points there is a diffusion junction. Where the fibers are loosely packed, the flow rate is high and a flowing junction exists.

The quartz fiber reference electrode consists of straight fibers of quartz laid next to each other with straight channels of filling solution passing between them. This junction generally is considered to have no diffusion properties and to be strictly a flowing junction.

The sleeve junction reference electrode is constructed by putting a hole in the side of a glass or plastic tube, grinding the surface around that tube, and covering it with a tapered glass or plastic sleeve. Such a junction is similar in its overall function to a frit junction in that certain areas are pressed tightly and other areas are not, so that both diffusion and flowing junctions exist. A sleeve junction electrode is much faster flowing and much easier to clean than the others.

Reference electrodes are available in a variety of types to accommodate many types of samples. Certain samples require reference electrodes that flow very slowly, whereas other samples require reference electrodes that are easy to clean. The flow chart in Figure 2 should serve as a guide in selecting the reference electrode most appropriate for the application.

The best way to understand the purpose of a reference electrode is to imagine measuring the voltage on a battery with a voltmeter. A voltage measurement cannot be made if only one end of the battery is connected to the meter. If, however, two leads are plugged into the voltmeter and both ends of the battery are touched, a reading is possible.

A pH electrode can be compared to the battery in this example. The wire inside the pH electrode serves the same purpose as the first lead from the voltmeter. The reference electrode can be compared to the second lead, which completes the circuit and enables the measurement of the voltage, or voltage changes, at the pH electrode.

In reality, the pH and reference electrodes are immersed in the same solution with the filling solution of the reference electrode flowing into the sample and completing the circuit with the pH electrode. If the reference electrode is not immersed in the sample, a legitimate measurement is not possible. If the reference electrode immersed in the sample is completely dry, no reading would be forthcoming. If it were filled with filling solution but completely occluded, then it would not establish good consistent electrical connection with the sample, and the readings would vary. A reference electrode that is partially plugged up allows the sample ions to migrate into the junction of the electrode and set up new potentials: voltages that are measured by the system and interpreted as changing pH readings. This is the most common source of problems in a pH measuring system.

The first indication of difficulty in the reference electrode usually is a very long stabilization time. This can be caused by changes in temperature, by reactions taking place in the solution, or by pickup of CO_2 from the atmosphere. Generally, however, a long stabilization time is caused by either the incompatibility of the reference electrode with the sample being measured or by a faulty reference electrode.

It is usually possible to differentiate between drift caused by a faulty electrode and drift caused by other factors (incompatibility between electrode and sample, temperature changes, or reaction within the sample) by moving a hand quickly toward and then away from the electrode. If the reading on the pH meter changes significantly in response to the hand movement, and if the change in the meter reading reverses when retracting the hand, then it would be very safe to assume that the reference electrode is either plugged up or otherwise defective. If the drift continues undisturbed by the movement of the hand toward and away from the electrode, then the problem is probably in the sample. Although this is not a foolproof method, it works most of the time.

There are other ways to check for improperly functioning reference electrodes. The most positive checks are performed using a magnetic stirrer.

- If stirring a sample seems to cause an unstable reading, turn the stirrer off. If the reading changes significantly (by one or two tenths of a pH unit), then there is a reference electrode problem.
- If, while stirring, there is a fair amount of noise (variation in the reading), and turning the stirring motor down to a slower speed reduces the amount of noise, it is safe to assume that there is a reference electrode problem.

If the reference electrode is so dirty that it is completely occluded, it may be just like an open circuit. If this is the case, it exhibits the typical slow, never-ending drift. There are a number of other sources of this same problem (including the electrode's not being plugged in to the meter!), such as: broken wires within the pH or reference electrode; a broken lead wire from the pH or reference electrode; or an open circuit within the meter. A quick way to check whether the problem is in the electrodes or the meter is to substitute new electrodes or, using a wire, paper clip, or shorting plug, short between the reference electrode input and the pH electrode input on the meter. If shorting the electrode inputs does

Z TECHNICAL REFERENCE

516

Reference Temperatures

We cannot build a temperature divider as we can a voltage divider. nor can we add temperatures as we would add lengths to measure distance. We must rely upon temperatures established by physical phenomena which are easily observed and consistent in nature. The International Practical Temperature Scale (IPTS) is based on such phenomena. Revised in 1968, it establishes eleven reference temperatures.

Since we have only these fixed temperatures to use as a reference, we must use instruments to interpolate between them. But accurately interpolating between these temperatures can require some fairly exotic transducers, many of which are too complicated or expensive to use in a practical situation. We shall limit our discussion to the four most common temperature transducers: thermocouples, resistance-temperature detector's (RTD's), thermistors, and integrated circuit sensors.

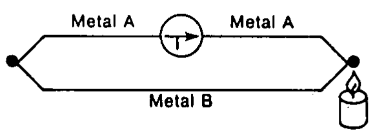
IPTS-68 REFERENCE TEMPERATURES

EQUILIBRIUM POINT	K	°C
Triple Point of Hydrogen	13.81	−259.34
Liquid/Vapor Phase of Hydrogen at 25/76 Std. Atmosphere	17.042	−256.108
Boiling Point of Hydrogen	20.28	−252.87
Boiling Point of Neon	27.102	−246.048
Triple Point of Oxygen	54.361	−218.789
Boiling Point of Oxygen	90.188	−182.962
Triple Point of Water	273.16	.01
Boiling Point of Water	373.15	100
Freezing Point of Zinc.	692.73	419.58
Freezing Point of Silver	1235.08	961.93
Freezing Point of Gold	1337.58	1064.43

Table 1

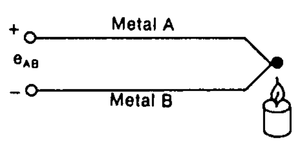
THE THERMOCOUPLE

When two wires composed of dissimilar metals are joined at both ends and one of the ends is heated, there is a continuous current which flows in the *thermoelectric* circuit. Thomas Seebeck made this discovery in 1821.



THE SEEBECK EFFECT
Figure 2

If this circuit is broken at the center, the net open circuit voltage (the Seebeck voltage) is a function of the junction temperature and the composition of the two metals.



e_{AB} = SEEBECK VOLTAGE
Figure 3

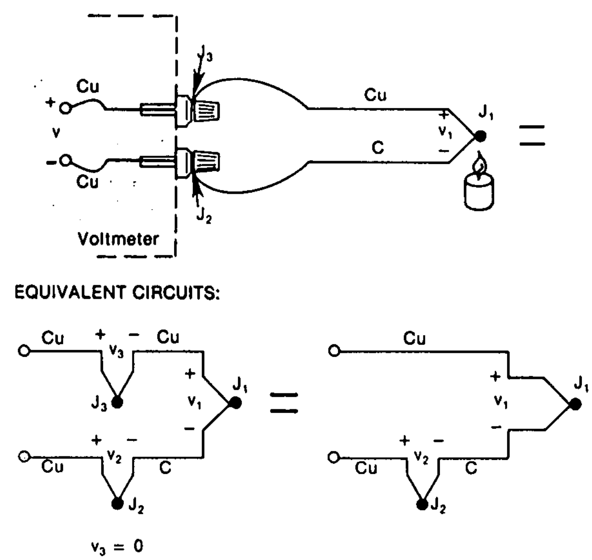
All dissimilar metals exhibit this effect. The most common combinations of two metals are listed in Appendix B of this application note, along with their important characteristics. For small changes in temperature the Seebeck voltage is linearly proportional to temperature:

$$\Delta e_{AB} = \alpha \Delta T$$

Where α, the Seebeck coefficient, is the constant of proportionality.

Measuring Thermocouple Voltage – We can't measure the Seebeck voltage directly because we must first connect a voltmeter to the thermocouple, and the voltmeter leads themselves create a new thermoelectric circuit.

Let's connect a voltmeter across a copper-constantan (Type T) thermocouple and look at the voltage output:

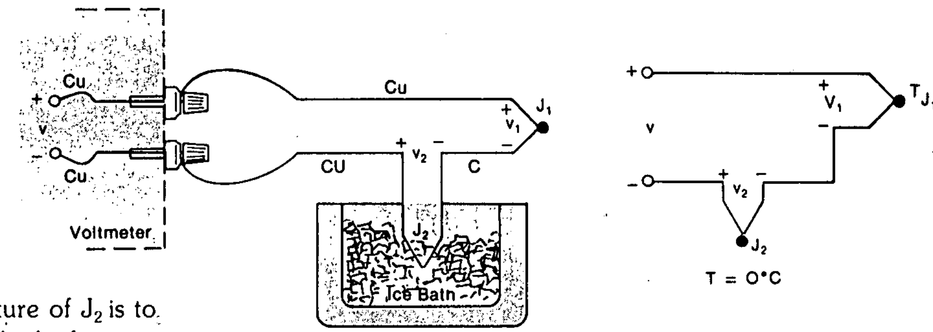


MEASURING JUNCTION VOLTAGE WITH A DVM
Figure 4

We would like the voltmeter to read only V_1, but by connecting the voltmeter in an attempt to measure the output of Junction J_1, we have created two more metallic junctions: J_2 and J_3. Since J_3 is a copper-to-copper junction, it creates no thermal EMF ($V_3 = 0$) but J_2 is a copper-to-constantan junction which will add an EMF (V_2) in opposition to V_1. The resultant voltmeter reading V will be proportional to the temperature *difference* between J_1 and J_2. This says that we can't find the temperature at J_1 unless we first find the temperature of J_2.

509

The Reference Junction



EXTERNAL REFERENCE JUNCTION
Figure 5

One way to determine the temperature of J_2 is to physically put the junction into an ice bath, forcing its temperature to be 0°C and establishing J_2 as the *Reference Junction*. Since both voltmeter terminal junctions are now copper-copper, they create no thermal emf and the reading V on the voltmeter is proportional to the temperature difference between J_1 and J_2.

Now the voltmeter reading is (See Figure 5):
$V = (V_1 - V_2) \cong \alpha\,(t_{J_1} - t_{J_2})$

If we specify T_{J_1} in degrees Celsius:
$T_{J_1}\,(°C) + 273.15 = t_{J_1}$

then V becomes:

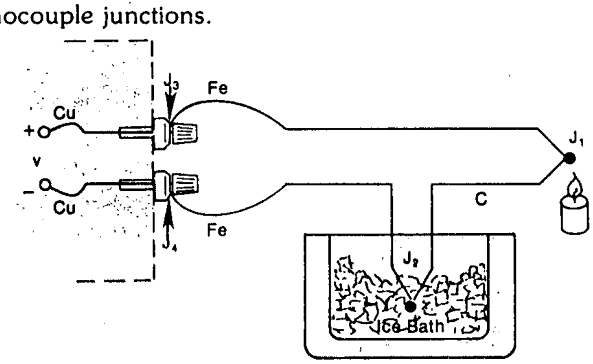
$$V = V_1 - V_2 = \alpha\left[(T_{J_1} + 273.15) - (T_{J_2} + 273.15)\right] = \alpha\,(T_{J_1} - T_{J_2}) = \alpha(T_{J_1} - 0)$$

$V = \alpha T_{J_1}$

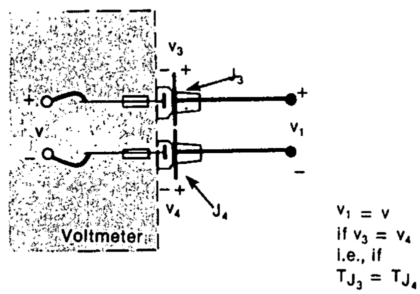
We use this protracted derivation to emphasize that the ice bath junction output, V_2, is *not* zero volts. It is a function of absolute temperature.

By adding the voltage of the ice point reference junction we have now referenced the reading V to 0°C. This method is very accurate because the ice point temperature can be precisely controlled. The ice point is used by the National Bureau of Standards (NBS) as the fundamental reference point for their thermocouple tables, so we can now look at the NBS tables and directly convert from voltage V to Temperature T_{J_1}.

The copper-constantan thermocouple shown in Figure 5 is a unique example because the copper wire is the same metal as the voltmeter terminals. Let's use an iron-constantan (Type J) thermocouple instead of the copper-constantan. The iron wire (Figure 6) increases the number of dissimilar metal junctions in the circuit, as both voltmeter terminals become Cu-Fe thermocouple junctions.

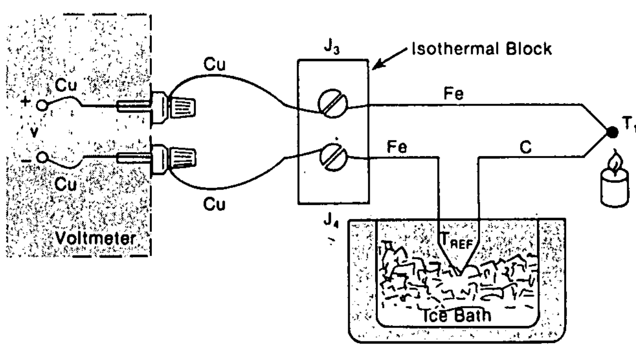


IRON-CONSTANTAN COUPLE
Figure 6



JUNCTION VOLTAGE CANCELLATION
Figure 7

If both front panel terminals are not at the same temperature, there will be an error. For a more precise measurement the copper voltmeter leads should be extended so the copper-to-iron junctions are made on an *isothermal* (same temperature) block:



REMOVING JUNCTIONS FROM DVM TERMINALS
Figure 8

The isothermal block is an electrical insulator but a good heat conductor and it serves to hold J_3 and J_4 at the same temperature. The absolute block temperature is unimportant because the two Cu-Fe junctions act in opposition. We still have

$$V = \alpha(T_1 - T_{REF})$$

Z

510

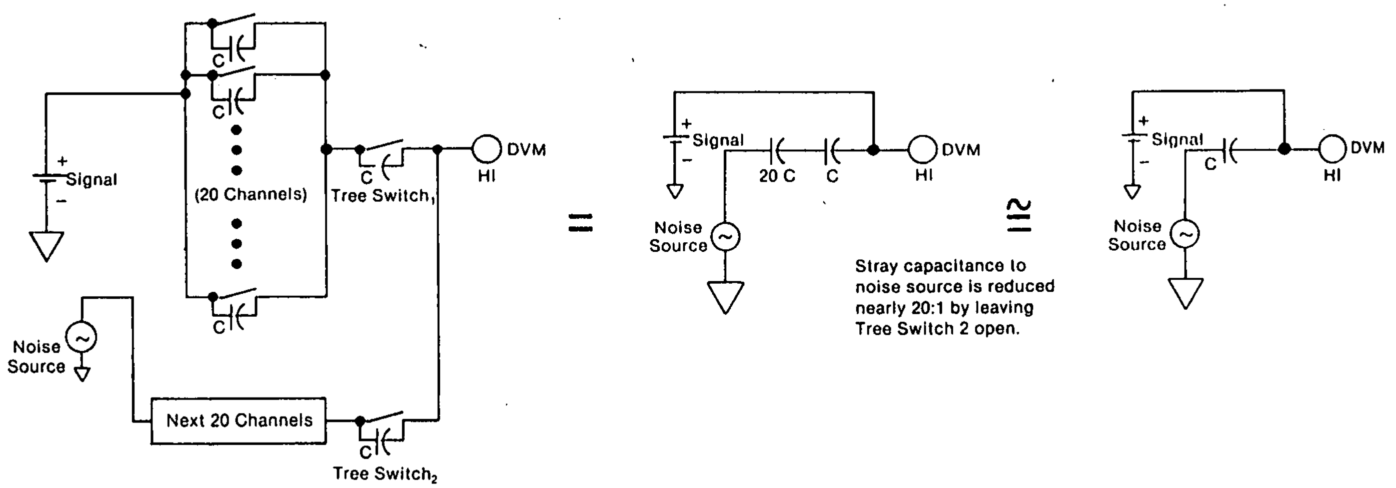
Noise Rejection

Tree Switching – Tree switching is a method of organizing the channels of a scanner into groups, each with its own main switch.

Without tree switching, every channel can contribute noise directly through its *stray* capacitance. With tree switching, groups of parallel channel capacitances are in series with a single *tree switch* capacitance. The result is greatly reduced crosstalk in a large data acquisition system, due to the reduced interchannel capacitance.

Guarding – Guarding is a technique used to reduce interference from any noise source that is common to both high and low measurement leads, i.e., from *common mode* noise sources.

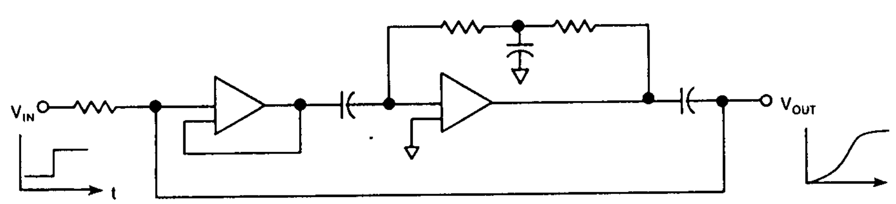
Let's assume a thermocouple wire has been pulled through the same conduit as a 220V AC supply line. The capacitance between the power lines and the thermocouple lines will create an AC signal of approximately equal magnitude on both thermocouple wires. This *common mode* signal is not a problem in an ideal



TREE SWITCHING
Figure 19

Analog Filter – A filter may be used directly at the input of a voltmeter to reduce noise. It reduces interference dramatically, but causes the voltmeter to respond more slowly to step inputs.
through the thermocouple lead resistance, creating a normal mode noise signal. The guard, physically a

circuit, but the voltmeter is not ideal. It has some capacitance between its low terminal and safety ground (chassis). Current flows through this capacitance and floating metal box surrounding the entire voltmeter circuit, is connected to a shield surrounding the thermocouple wire, and serves to shunt the interfering current.



ANALOG FILTER
Figure 20

Integration – Integration is an A/D technique which essentially averages noise over a full line cycle, thus power line-related noise and its harmonics are virtually eliminated. If the integration period is chosen to be less than an integer line cycle, its noise rejection properties are essentially negated.

Since thermocouple circuits that cover long distances are especially susceptible to power line related noise, it is advisable to use an integrating analog to digital conveter to measure the thermocouple voltage. Integration is an especially attractive A/D technique in light of recent innovations which allow reading rates of 48 samples per second with full cycle integration.
through the thermocouple lead resistance, creating a normal mode noise signal. The guard, physically a floating metal box surrounding the entire voltmeter circuit, is connected to a shield surrounding the thermocouple wire, and serves to shunt the interfering current.

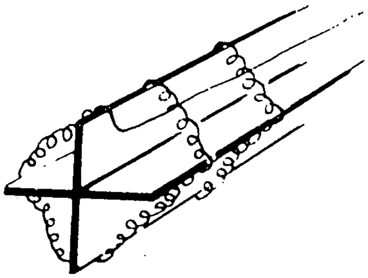
THE RTD

History

The same year that Seebeck made his discovery about thermoelectricity, Sir Humphrey Davy announced that the resistivity of metals showed a marked temperature dependence. Fifty years later, Sir William Siemens proffered the use of platinum as the element in a resistance thermometer. His choice proved most propitious, as platinum is used to this day as the primary element in all high-accuracy resistance thermometers. In fact, the Platinum Resistance Temperature Detector, or PRTD, is used today as an interpolation standard from the oxygen point (− 182.96°C) to the antimony point (630.74°C).

Platinum is especially suited to this purpose, as it can withstand high temperatures while maintaining excellent stability. As a noble metal, it shows limited susceptibility to contamination.

The classical resistance temperature detector (RTD) construction using platinum was proposed by C.H. Meyers in 1932.[12] He wound a helical coil of platinum on a crossed mica web and mounted the assembly inside a glass tube. This construction minimized strain on the wire while maximizing resistance.



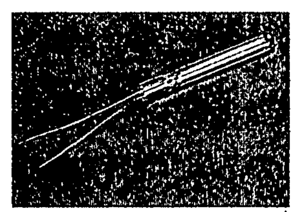
MEYERS RTD CONSTRUCTION
Figure 35

Although this construction produces a very stable element, the thermal contact between the platinum and the measured point is quite poor. This results in a slow thermal response time. The fragility of the structure limits its use today primarily to that of a laboratory standard.

Another laboratory standard has taken the place of the Meyer's design. This is the *bird-cage* element proposed by Evans and Burns.[16] The platinum element remains largely unsupported, which allows it to move freely when expanded or contracted by temperature variations. Strain-induced resistance changes with time and temperature are thus minimized, and the bird-cage becomes the ultimate laboratory standard. Due to the unsupported structure and subsequent susceptibility to vibration, this configuration is still a bit too fragile for industrial environments.

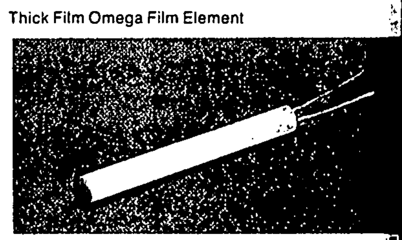
A more rugged construction technique is shown in Figure 37. The platinum wire is bifilar wound on a glass or ceramic bobbin. The bifilar winding reduces the effective enclosed area of the coil to minimize magnetic pickup and its related noise. Once the wire is wound onto the bobbin, the assembly is then sealed with a coating of molten glass. The sealing process assures that the RTD will maintain its integrity under extreme vibration, but it also limits the expansion of the platinum metal at high temperatures. Unless the coefficients of expansion of the platinum and the bobbin match perfectly, stress will be placed on the wire as the temperature changes, resulting in a strain-induced resistance change. This may result in a permanent change in the resistance of the wire.

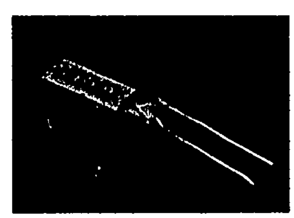
There are partially supported versions of the RTD which offer a compromise between the bird-cage approach and the sealed helix. One such approach uses a platinum helix threaded through a ceramic cylinder and affixed via glass-frit. These devices will maintain excellent stability in moderately rugged vibrational applications.



Glass Sealed Bifilar Winding

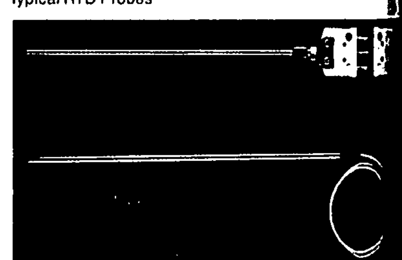
Thick Film Omega Film Element





Thin Film Omega TFD Element

Typical RTD Probes



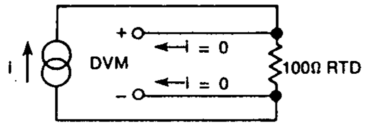
TYPICAL RTDs
Figures 36 and 37

Metal Film RTD's

In the newest construction technique, a platinum or metal-glass slurry film is deposited or screened onto a small flat ceramic substrate, etched with a laser-trimming system, and sealed. The film RTD offers substantial reduction in assembly time and has the further advantage of increased resistance for a given size. Due to the manufacturing technology, the device size itself is small, which means it can respond quickly to step changes in temperature. Film RTD's are presently less stable than their hand-made counterparts, but they

12 Refer to Bibliography 12.
16 Refer to Bibliography 16.

4-Wire Ohms – The technique of using a current source along with a remotely sensed digital voltmeter alleviates many problems associated with the bridge.



4-WIRE OHMS MEASUREMENT
Figure 42

The output voltage read by the dvm is directly proportional to RTD resistance, so only one conversion equation is necessary. The three bridge-completion resistors are replaced by one reference resistor. The digital voltmeter measures only the voltage dropped across the RTD and is insensitive to the length of the lead wires.

The one disadvantage of using 4-wire ohms is that we need one more extension wire than the 3-wire bridge. This is a small price to pay if we are at all concerned with the accuracy of the temperature measurement.

3-Wire Bridge Measurement Errors

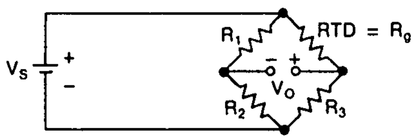


Figure 43

If we know V_S and V_O, we can find R_g and then solve for temperature. The unbalance voltage V_O of a bridge built with $R_1 = R_2$ is:

$$V_O = V_S\left(\frac{R_3}{R_3 + R_g}\right) - V_S\left(\frac{1}{2}\right)$$

If $R_g = R_3$, $V_O = 0$ and the bridge is balanced. This can be done manually, but if we don't want to do a manual bridge balance we can just solve for R_g in terms of V_O:

$$R_g = R_3\left(\frac{V_S - 2V_O}{V_S + 2V_O}\right)$$

This expression assumes the lead resistance is zero. If R_g is located some distance from the bridge in a 3-wire configuration, the lead resistance R_L will appear in series with both R_g and R_3:

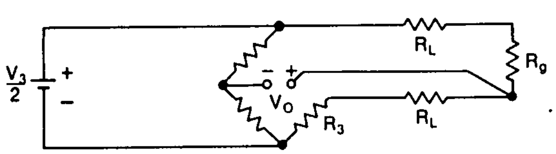


Figure 44

Again we solve for R_g:

$$R_g = R_3\left(\frac{V_S - 2V_O}{V_S + 2V_O}\right) - R_L\left(\frac{4V_O}{V_S + 2V_O}\right)$$

The error term will be small if V_O is small, i.e., the bridge is close to balance. This circuit works well with devices like strain gauges, which change resistance value by only a few percent, but an RTD changes resistance dramatically with temperature. Assume the RTD resistance is 200 ohms and the bridge is designed for 100 ohms:

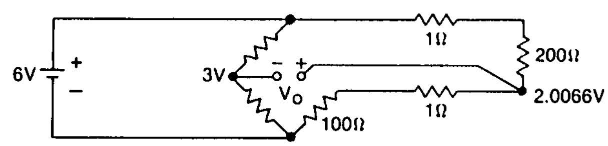


Figure 45

Since we don't know the value of R_L, we must use equation (a), so we get:

$$R_g = 100\left(\frac{6 - 1.9868}{6 + 1.9868}\right) = 199.01 \text{ ohms}$$

The correct answer is of course 200 ohms. That's a temperature error of about 2½°C.

Unless you can actually measure the resistance of R_L or balance the bridge, the basic 3-wire technique is not an accurate method for measuring absolute temperature with an RTD. A better approach is to use a 4-wire technique.

Resistance to Temperature Conversion

The RTD is a more linear device than the thermocouple, but it still requires curve-fitting. The Callendar-Van Dusen equation has been used for years to approximate the RTD curve:[11, 13]

$$R_T = R_0 + R_0\,\alpha\left[T - \delta\left(\frac{T}{100} - 1\right)\left(\frac{T}{100}\right) - \beta\left(\frac{T}{100} - 1\right)\left(\frac{T^3}{100}\right)\right]$$

Where:

R_T = Resistance at Temperature T
R_O = Resistance at T = 0°C
α = Temperature coefficient at T = 0°C (typically +0.00392Ω/Ω/°C)
δ = 1.49 (typical value for .00392 platinum)
β = 0   T > 0
0.11 (typical) T < 0

The exact values for coefficients α, β, and δ are determined by testing the RTD at four temperatures and solving the resultant equations. This familiar equation was replaced in 1968 by a 20th order polynomial

[11, 13] Refer to Bibliography 11 and 13.